

Titulació:

Grau en Enginyeria Elèctrica

Alumne (nom i cognoms):

Alfocea Jimenez, Marc

Enunciat TFG:

CONTROL Y SUPERVISIÓN DE UN VARIADOR DE VELOCIDAD CON UN
AUTÓMATA Y UN SISTEMA HMI

Director/a del TFG / TFM:

Romero Duran, David

Convocatòria de lliurament del TFG / TFM:

Maig 2019

Abstract

[Castellano]

Este proyecto diseña un sistema de control y supervisión de un motor, a partir de tecnología de ABB. El control se realiza mediante un autómatas programable y un panel HMI, que actúan sobre un convertidor. Las comunicaciones del sistema se ejecutan mediante el protocolo estándar, Modbus RTU. El sistema se desarrolla a partir de las herramientas de software de ABB basados en el estándar IEC 61131-3 y se reproduce, a modo de manual, todo el proceso de creación y explicación del entorno.

El control realizado proporciona la posibilidad de actuar sobre el motor a velocidades variables y permitir la parametrización del convertidor. Asimismo, el sistema de supervisión permite controlar en tiempo real el estado del sistema, así como, los parámetros más significativos del convertidor.

[English]

This project designs a system for the control and supervision of an engine, based on ABB technology. The control is carried out by means of a programmable automaton and an HMI panel that act on a converter. System communications are executed using the standard protocol, Modbus RTU. The system is developed from the ABB software tools based on the IEC 61131-3 standard and reproduces, by like a form of manual, the whole process of creation and explanation of the environment.

The control carried out provides the possibility of acting on the motor at variable speeds and allowing parameterization of the converter. Likewise, the supervision system allows to control in real time the state of the system, as well as, the most significant parameters of the converter.

Agradecimientos

Quiero agradecerle a mi profesor y tutor, David Romero Duran, sus consejos técnicos, sus ánimos y su comprensión. Por entender las dificultades de los que tenemos otras responsabilidades, profesionales y familiares, que no podemos desatender. Pero sobretodo, por transmitirme la vocación por la programación y la automatización, que me acompañará a lo largo de toda mi etapa profesional.

Dedicatoria

Quiero dedicar este trabajo a mi madre Juani y a mi pareja Nuria, por compartir mis esfuerzos e ilusiones. Porque sin vosotras este trabajo no habría sido posible.

A ti, mama, por estar siempre ahí y porque eres el mayor ejemplo de lucha y perseverancia que conozco, eres imprescindible.

“Hay hombres que luchan un día y son buenos. Hay otros que luchan un año y son mejores. Hay quienes luchan muchos años, y son muy buenos. Pero los hay que luchan toda la vida: esos son los imprescindibles “. Bertolt Brecht.

A ti, Nuria, por caminar conmigo, por devolverme la esperanza, por tu natural sencillez, por compartir mis sueños, por ser mi sueño.

“Las palabras nunca alcanzan cuando lo que hay que decir desborda el alma”. Julio Cortázar

Índice

Abstract.....	1
Índice de tablas.....	6
Índice de figuras.....	10
Glosario.....	10
1. Introducción	12
2. Antecedentes	13
3. Alcance y objeto del proyecto	14
4. Metodología	15
5. Descripción del sistema.....	16
6. Hardware	17
6.1. Convertidor de frecuencia.....	17
6.1.1. Fundamentos teóricos.....	17
6.1.2. Modelo: ABB ACS550.....	20
6.1.3. Características técnicas.....	21
6.1.4. Macros y configuración de parámetros.....	24
6.1.5. Sistemáticas de funcionamiento	27
6.2. PAC	30
6.2.1. Fundamentos teóricos.....	30
6.2.2. Autómatas programables: ABB.....	32
6.2.3. Terminal base ABB TB511	34
6.2.4. PAC ABB PM573 – ETH.....	34
6.2.5. ABB DA501	36
6.2.6. ABB CD522	38
6.2.7. Configuración física del autómata	41
6.3. HMI.....	42
6.3.1. Fundamentos teóricos.....	42
6.3.2. Modelo: ABB CP630	43
6.3.3. Características técnicas.....	44
6.4. Instrucciones de conexionado.....	47
6.4.1. Conexión PC – PAC.....	47
6.4.2. Conexión PAC – Variador de frecuencia	47
6.4.3. Conexión Panel HMI-PAC.....	49

6.4.4.	Cableado del sistema.....	50
7.	Software	52
7.1.	Normalización IEC61131-3	52
7.2.	Automation Builder: Suite integral para la automatización.....	54
7.2.1.	Instalación.....	54
7.2.2.	Explicación del entorno	56
7.2.3.	Creación de un nuevo proyecto.....	59
7.2.4.	Configuración del autómata	63
7.2.5.	Configuración IP.....	65
7.2.6.	Asignación de variables	67
7.2.7.	Bibliotecas.....	69
7.2.8.	Actualización firmware autómata	71
7.2.9.	Errores software	74
7.2.9.1.	Borrar errores del autómata.....	74
7.2.9.2.	Error batería auxiliar.....	75
7.2.9.3.	Tiempo de espera actualizar datos.....	76
7.3.	CODESYS: plataforma de programación.....	78
7.3.1.	Explicación del entorno	79
7.3.2.	Lenguajes de programación.....	81
7.3.3.	Definición de variables	84
7.3.4.	Programación en FBD	88
7.3.5.	Visualizaciones.....	98
7.3.6.	Modo online y modo simulación	107
7.4.	Panel Builder 600	110
7.4.1.	Explicación del entorno	111
7.4.2.	Creación de un proyecto nuevo.....	113
7.4.3.	Importación de variables	115
7.4.4.	Programación y diseño	118
7.4.5.	Descargar el programa	128
8.	Comunicación	132
8.1.	Modbus RTU	132
8.2.	Configuración convertidor	134
8.3.	Configuración PAC.....	136

8.4.	Configuración HMI	140
9.	Control y supervisión de un convertidor mediante PAC y HMI.....	142
9.1.	Programa y visualización PAC.....	142
9.2.	HMI: Interfaz de usuario	153
10.	Conclusiones.....	157
11.	Bibliografía.....	158

Índice de Figuras

Figura 1- Arquitectura general del sistema	16
Figura 2 – Diagrama de bloques de un convertidor indirecto [30]	17
Figura 3 – Tipos de rectificador [32]	18
Figura 4 – Inversor PWM [32]	19
Figura 5 – Vista general convertidor ACS550 [30].....	20
Figura 6 – Esquema de conexiones convertidor ACS550 [30].....	20
Figura 7 – Funcionalidades panel ACS550 [30]	21
Figura 8 – Control ACS550 mediante E/S del convertidor [30]	29
Figura 9 – Diagrama básico de un PAC.....	30
Figura 10 - Autómata escalable AC500 [27]	32
Figura 11 – Vista general Base de Terminales TB511 [19]	34
Figura 12 – Vista general autómata PM573-ETH [29]	35
Figura 13 – Funcionalidades panel PM573-ETH [29]	35
Figura 14 – Vista general módulo DA501 [22]	37
Figura 15 - Funcionalidades módulo DA501 [22]	37
Figura 16 – Vista general del módulo CD522 [22].....	38
Figura 17 - Funcionalidades módulo CD522 [22]	39
Figura 18 – Configuración del autómata	41
Figura 19 – Vista general del panel CP630 [15].....	44
Figura 20 – Conexiones del panel CP630 [15]	44
Figura 21 – Pins de conexionado COM1 PM573-ETH [28]	48
Figura 22 – Conexión RS485 PAC PM573-ETH-Convertidor ACS550 [28] [30]	48
Figura 23 – Conmutador RS485 ACS550 [30]	49
Figura 24 – Cable conexionado PM573-ETH COM2 – CP630	49
Figura 25 – Pins conexionado RS485 CP630 [28]	49
Figura 26 – Cable con terminales DE-9 conexión PM573-ETH – CP630 [12]	50
Figura 27 – Conexionado de alimentación módulos DA501 – CD522 [20] [25]	50
Figura 28 – Conexionado de alimentación del sistema.....	51
Figura 29 – Automation Builder. Proceso de Instalación 1	55
Figura 30 - Automation Builder. Proceso de Instalación 2.....	55
Figura 31 - Automation Builder. Proceso de Instalación 3.....	56
Figura 32 – Automation Builder. Entorno	59
Figura 33 – Automation Builder. Creación nuevo proyecto 1.....	60
Figura 34 - Automation Builder. Creación nuevo proyecto 2	60
Figura 35 - Automation Builder. Creación nuevo proyecto 3	61
Figura 36 - Automation Builder. Creación nuevo proyecto 4	61
Figura 37 - Automation Builder. Creación nuevo proyecto 5	62
Figura 38 - Automation Builder. Creación nuevo proyecto 6	62
Figura 39 - Automation Builder. Creación nuevo proyecto 7	63
Figura 40 – Automation Builder. Configuración autómata 1	64
Figura 41 - Automation Builder. Configuración autómata 2.....	64

Figura 42 - Automation Builder. Configuración automática 3	65
Figura 43 – Automation Builder. Configuración IP 1	66
Figura 44 - Automation Builder. Configuración IP 2	66
Figura 45 - Automation Builder. Configuración IP 3	67
Figura 46 - Automation Builder. Asignación de variables 1	68
Figura 47 - Automation Builder. Asignación de variables 2	68
Figura 48 - Automation Builder. Asignación de variables 3	69
Figura 49 - Automation Builder. Instalación de bibliotecas paso 1	70
Figura 50 - Automation Builder. Instalación de bibliotecas paso 2	71
Figura 51 - Automation Builder. Actualización firmware paso 1	72
Figura 52 - Automation Builder. Actualización firmware paso 2	72
Figura 53 - Automation Builder. Actualización firmware paso 3	73
Figura 54 - Automation Builder. Actualización firmware paso 4	73
Figura 55 - Automation Builder. Borrar errores del autómeta	75
Figura 56 - Automation Builder. Error batería auxiliar 1	76
Figura 57 - Automation Builder. Error batería auxiliar 2	76
Figura 58 - Automation Builder. Error tiempo de espera actualizar datos 1	77
Figura 59 - Automation Builder. Error tiempo de espera actualizar datos 2	77
Figura 60 – Icono CODESYS	78
Figura 61 – CODESYS. Entorno	80
Figura 62 - CODESYS. Lenguajes de programación estándar (IL) [10]	81
Figura 63 - CODESYS. Lenguajes de programación estándar (ST) [10]	82
Figura 64 - CODESYS. Lenguajes de programación estándar (FBD) [10]	82
Figura 65 - CODESYS. Lenguajes de programación estándar (LD) [10]	83
Figura 66 - CODESYS. Lenguajes de programación estándar (SFC) [10]	83
Figura 67 - CODESYS. Lenguajes de programación estándar (CFC) [10]	84
Figura 68 - CODESYS. Declaración de variables 1	85
Figura 69 - CODESYS. Declaración de variables 2	86
Figura 70 - CODESYS. Declaración de variables 3	86
Figura 71 - CODESYS. Declaración de variables 4	87
Figura 72 - CODESYS. Declaración de variables 5	87
Figura 73 - CODESYS. Programación en FBD 1	89
Figura 74 - CODESYS. Programación en FBD 2	90
Figura 75 - CODESYS. Programación en FBD 3	90
Figura 76 - CODESYS. Programación en FBD 4	91
Figura 77 - CODESYS. Programación en FBD 5	92
Figura 78 - CODESYS. Programación en FBD 6	93
Figura 79 - CODESYS. Programación en FBD 7	95
Figura 80 - CODESYS. Programación en FBD 8	96
Figura 81 - CODESYS. Programación en FBD 9	97
Figura 82 - CODESYS. Programación en FBD 10	98
Figura 83 – CODESYS. Creación de visualizaciones 1	99
Figura 84 - CODESYS. Creación de visualizaciones 2	99
Figura 85 - CODESYS. Creación de visualizaciones 3	99

Figura 86 - CODESYS. Creación de visualizaciones 4	100
Figura 87 - CODESYS. Creación de visualizaciones 5	100
Figura 88 - CODESYS. Creación de visualizaciones 6	101
Figura 89 - CODESYS. Creación de visualizaciones 7	101
Figura 90 - CODESYS. Creación de visualizaciones 8	102
Figura 91 - CODESYS. Creación de visualizaciones 9	103
Figura 92 - CODESYS. Creación de visualizaciones 10	103
Figura 93 - CODESYS. Creación de visualizaciones 11	104
Figura 94 - CODESYS. Creación de visualizaciones 12	104
Figura 95 - CODESYS. Creación de visualizaciones 13	105
Figura 96 - CODESYS. Creación de visualizaciones 14	105
Figura 97 - CODESYS. Creación de visualizaciones 15	106
Figura 98 - CODESYS. Creación de visualizaciones 16	107
Figura 99 - CODESYS. Modo simulación 1	107
Figura 100 - CODESYS. Modo simulación 2	108
Figura 101 - CODESYS. Modo simulación 3	108
Figura 102 - CODESYS. Modo simulación 4	109
Figura 103 - CODESYS. Modo simulación 5	109
Figura 104 – Panel Builder. Entorno.....	112
Figura 105 – Panel Builder. Creación nuevo proyecto 1	113
Figura 106 - Panel Builder. Creación nuevo proyecto 2.....	114
Figura 107 - Panel Builder. Creación nuevo proyecto 3.....	114
Figura 108 - Panel Builder. Importación de variables 1	115
Figura 109 - Panel Builder. Importación de variables 2	116
Figura 110 - Panel Builder. Importación de variables 3	116
Figura 111 - Panel Builder. Importación de variables 4	117
Figura 112 - Panel Builder. Importación de variables 5	117
Figura 113 - Panel Builder. Programación 1.....	118
Figura 114 - Panel Builder. Programación 2.....	119
Figura 115 - Panel Builder. Programación 3.....	120
Figura 116 - Panel Builder. Programación 4.....	120
Figura 117 - Panel Builder. Programación 5.....	121
Figura 118 - Panel Builder. Programación 6.....	121
Figura 119 - Panel Builder. Programación 7	122
Figura 120 - Panel Builder. Programación 8.....	122
Figura 121 - Panel Builder. Programación 9.....	123
Figura 122 - Panel Builder. Programación 10.....	123
Figura 123 - Panel Builder. Programación 11.....	124
Figura 124 - Panel Builder. Programación 12.....	124
Figura 125 - Panel Builder. Programación 13.....	125
Figura 126 - Panel Builder. Programación 14.....	125
Figura 127 - Panel Builder. Programación 15.....	126
Figura 128 - Panel Builder. Programación 16.....	127
Figura 129 - Panel Builder. Programación 17	128

Figura 130 - Panel Builder. Descargar programa en panel CP630 1.....	128
Figura 131 - Panel Builder. Descargar programa en panel CP630 2.....	129
Figura 132 - Panel Builder. Descargar programa en panel CP630 3.....	129
Figura 133 - Panel Builder. Descargar programa en panel CP630 4.....	130
Figura 134 - Panel Builder. Descargar programa en panel CP630 5.....	130
Figura 135 - Panel Builder. Descargar programa en panel CP630 6.....	131
Figura 136 – Modbus RTU. Formato General de tramas [23]	133
Figura 137 - Modbus RTU. Configuración Automation Builder convertidor 1	137
Figura 138 - Modbus RTU. Configuración Automation Builder convertidor 2	138
Figura 139 - Modbus RTU. Configuración Automation Builder panel HMI 1	138
Figura 140 - Modbus RTU. Configuración Automation Builder panel HMI 2	139
Figura 141 - Modbus RTU. Configuración Automation Builder panel HMI 3	139
Figura 142 - Modbus RTU. Configuración Panel Builder panel HMI 1	140
Figura 143 - Modbus RTU. Configuración Panel Builder panel HMI 2	141
Figura 144 - Modbus RTU. Configuración Panel Builder panel HMI 3	141
Figura 145 – Automation Builder: Programa PAC control y supervisión 1	142
Figura 146 - Automation Builder: Programa PAC control y supervisión 2	143
Figura 147 - Automation Builder: Programa PAC control y supervisión 3	143
Figura 148 - Automation Builder: Programa PAC control y supervisión 4	144
Figura 149 - Automation Builder: Programa PAC control y supervisión 5	144
Figura 150 - Automation Builder: Programa PAC control y supervisión 6	145
Figura 151 - Automation Builder: Programa PAC control y supervisión 7	145
Figura 152 - Automation Builder: Programa PAC control y supervisión 8	146
Figura 153 - Automation Builder: Programa PAC control y supervisión 9	146
Figura 154 - Automation Builder: Programa PAC control y supervisión 10	147
Figura 155 - Automation Builder: Programa PAC control y supervisión 11	148
Figura 156 - Automation Builder: Programa PAC control y supervisión 12	149
Figura 157 - Automation Builder: Programa PAC control y supervisión 13	150
Figura 158 - Automation Builder: Programa PAC control y supervisión 14	151
Figura 159 - Automation Builder: Programa PAC control y supervisión 15	151
Figura 160 - Automation Builder: Programa PAC control y supervisión 16	152
Figura 161 - Automation Builder: Programa PAC control y supervisión. Visualización	152
Figura 162 – Panel Builder: Programa control y supervisión HMI 1	153
Figura 163 - Panel Builder: Programa control y supervisión HMI 2	154
Figura 164 - Panel Builder: Programa control y supervisión HMI 3	154
Figura 165 - Panel Builder: Programa control y supervisión HMI 4	155
Figura 166 - Panel Builder: Programa control y supervisión HMI 5	155
Figura 167 - Panel Builder: Programa control y supervisión HMI 6	156
Figura 168 - Panel Builder: Programa control y supervisión HMI 7	156

**[...]Referencia bibliográfica de las figuras que no son de elaboración propia.*

Índice de tablas

Tabla 1 – Características técnicas convertidor ACS550.....	24
Tabla 2 – Macros de aplicación ACS550.....	25
Tabla 3 – Grupo de parámetros ACS550	27
Tabla 4 – Características técnicas autómata PM573-ETH	36
Tabla 5 – Características técnicas módulo DA501	38
Tabla 6 - Características técnicas módulo CD522	40
Tabla 7 - Características técnicas panel CP630	46
Tabla 8 – Configuración IP's comunicación PC /PAC.....	65
Tabla 9 – Configuración E/S del bloque funcional ACS_COM_MOD_RTU	93
Tabla 10 - Configuración E/S del bloque funcional ACS_DRIVES_CTRL_STANDARD.....	95
Tabla 11 - Configuración E/S del bloque funcional ACS_MOD_READ_N_PRM.....	95
Tabla 12 - Configuración E/S del bloque funcional ACS_MOD_WRITE_N_PRM.....	96
Tabla 13 – Parámetros básicos ACS550 para la comunicación Modbus RTU	135
Tabla 14 – Parámetros adicionales ACS550 para la comunicación Modbus RTU	136

Glosario

ABB - Asea Brown Boveri. Empresa especializada en automatización y robótica

BCI – Bus de campo encajado

CA/AC – Corriente alterna / Alternating current

CC/DC – Corriente continua / Direct current

CPU – Central processing unit – Unidad central de procesamiento

EA – Entrada analógica

ED – Entrada digital

EMC – Electromagnetic compatibility – Compatibilidad electromagnética

EPROM - Erasable programmable read-only memory - Memoria de solo lectura borrrable programable

ERP - Enterprise resource planning – Sistema de planificación de recursos empresariales

E/S – Entradas / Salidas

FBD - Function block diagram – Diagrama de bloques funcionales

FTP - File transfer protocol – Protocolo de transferencia de archivos

HMI – Human machine interface – Interfaz de usuario

IEC - International Electrotechnical Commission – Comisión electrotécnica internacional

IGBT - Insulated gate bipolar transistor – Transistor bipolar de puerta aislada

IoT - Internet of things – Internet de las cosas

IP – Internet protocol – Protocolo de internet

IPXX – Ingress protection – Índice de protección

LED - Light emitting diode – Diodo emisor de luz

- OPC** - OLE for Process Control – Vinculación e incrustación de objetos para procesos de control
- OSI** - Open system interconnection – Interconexión de sistemas abiertos
- PAC** - Programmable automation controller – Controlador de automatización programable
- PC** – Personal computer – Ordenador personal
- PID** - Controlador proporcional, integral y derivativo
- PLC** - Programmable logic controller – Controlador lógico programable
- POU** – Program organization unit – Unidad de organización del programa
- PWM** - Pulse-width modulation – Modulación por ancho de pulsos
- RAM** - Random access memory – Memoria de acceso aleatorio
- RTC** – Real time clock – Reloj en tiempo real
- RTU** – Unidad terminal remota
- SCADA** - Supervisory control and data acquisition - Supervisión, Control y Adquisición de Datos
- SVG** - Scalable vector graphics – Gráficos vectoriales escalables
- TCP** - Transmission control protocol – Protocolo de control de transmisión
- UNE** – Una norma española
- USB** - Universal serial bus – Bus universal en serie
- WMF** – Windows metafile – Meta-archivo de windows

1. Introducció

Un planeta de recursos limitados obliga a replantearnos el paradigma energètic. La energia fòsil, motor del desenvolupament humà fins a l'actualitat, és escassa i comporta danys que no podem externalitzar. En aquest sentit, en els pròxims anys nos enfrontem a un repte econòmic i social que va transformar la nostra realitat. Les iniciatives han de produir-se en diversos àmbits i disciplines diferents, si volem resoldre el problema com una història de èxit.

La consecució d'un model sostenible implica l'actuació en camps com el consum. Hem de replantearnos com utilitzem els recursos, per aquest motiu, l'eficiència és un element clau en aquest entramat. Per aconseguir límits acceptables, l'automatització i supervisió dels sistemes és essencial.

En aquest projecte es presenta un sistema de control i supervisió d'un motor, a partir de tecnologia de ABB. El control es realitza mitjançant un autòmat programable i un panel HMI que actua sobre un convertidor. Les comunicacions del sistema es executen mitjançant el protocol públic Modbus RTU. El sistema es desenvolupa a partir de les eines de software de ABB basades en l'estàndard IEC 61131 i es reprodueix tot el procés de creació i explicació del entorn.

L'automatització nos ajuda a consumir la energia elèctrica de manera eficient. En canvi, amb la supervisió dels sistemes ho fem de manera conscient. És dir, el coneixement del sistema nos permet actuar per millorar els processos i, per tant, reduir el consum energètic.

Per altra banda, com enginyer entenc la tecnologia com l'eina per millorar la vida de les persones. Sento gran passió i entusiasme per com el desenvolupament de la tècnica i la seva aplicació pot influir positivament en el benestar de les persones. És per tot això que es presenta el següent projecte, amb la finalitat de créixer com enginyer i contribuir, mínimament, a la realització d'un món millor.

2. Antecedentes

La automatización y las nuevas tecnologías en la industria son cada vez más un requisito indispensable para competir en el mercado. La velocidad de cambio e innovación en la actualidad tiene un ritmo exponencial.

Estos cambios tan rápidos han hecho aparecer tecnologías no estandarizadas o poco compatibles entre sí. La empresa ABB, entre otras, ha apostado por un entorno de programación y diseño de sistemas automatizados basado en los estándares IEC61113.

La regulación y estandarización implica fomentar un conocimiento válido para diferentes tecnologías y fabricantes. Además de ser un ahorro económico, proporciona una fácil adaptación profesional y flexibilización de los conocimientos. Aportando así, la posibilidad de democratizar el conocimiento, permitiendo participar en diferentes entornos y ubicaciones geográficas de manera sencilla y rápida.

Este proyecto intenta profundizar en el conocimiento de esta tecnología estándar, con la intención de adquirir unas competencias universales en la materia, y por tanto, aumentar las capacidades profesionales más allá de las restricciones que proporcionan el software propietario de un único fabricante.

3. Alcance y objeto del proyecto

El proyecto tiene la misión de crear una aplicación de control y supervisión de un motor a través de un convertidor. El control debe realizarse mediante autómeta programable y panel HMI. La comunicación entre los distintos dispositivos se realiza a través del protocolo de comunicación Modbus RTU.

El control del motor debe proporcionar la posibilidad de actuar a velocidades variables y permitir la parametrización del convertidor. Queda fuera del alcance del proyecto, el estudio y posibilidades de parametrización del convertidor.

El sistema de supervisión debe permitir controlar en tiempo real el estado del sistema, así como, los parámetros más significativos del convertidor.

Objetivos

- Revisión bibliográfica y posterior análisis de la documentación seleccionada, con el fin de, estudiar los principios de la tecnología y los dispositivos empleados.
- Estudio en profundidad de las diferentes herramientas de softwares empleadas para el desarrollo de la aplicación: Automation Builder, CODESYS y Panel Builder.
- Creación de un programa de control y supervisión controlado por un autómeta programable y un panel HMI con velocidades variables.
- Conocer y poner en práctica los lenguajes de programación del estándar IEC6113-3.
- Adquirir el conocimiento sobre el protocolo estándar Modbus RTU y realizar la comunicación de los dispositivos mediante dicho protocolo.
- Realizar un sistema HMI eficiente, eficaz y amigable.
- Creación de una simulación práctica que compruebe el programa y realizar así, su posterior depuración para conseguir el correcto funcionamiento de la aplicación.
- Realizar un informe del proyecto detallando el trabajo realizado y las tecnologías utilizadas.

- Explicitar de manera pràctica, a modo de manual, todo el proceso de creación. Para obtener un documento de referencia que pueda ser utilizado posteriormente, para la creación de nuevos proyectos que profundicen en esta tecnología utilizando las mismas herramientas.

4. Metodología

El método de trabajo para llevar a cabo el presente proyecto de desarrollo, se ha basado en una parte inicial de carácter cualitativo y una parte posterior de origen, fundamentalmente, cuantitativo.

La parte cualitativa ha tenido el objetivo de formarme y adquirir las herramientas necesarias para el desarrollo de la parte práctica. Basada, principalmente, en la revisión bibliográfica. A partir de la cual, se ha generado un proceso de recogida y análisis de la información existente. Estos conocimientos han sido integrados en la capacitación adquirida durante el grado y mi actividad profesional, que me han permitido adaptarme a una tecnología diferente a partir de los conocimientos generales de una tecnología concreta.

La parte cuantitativa corresponde al desarrollo práctico del sistema control. Esta etapa del proyecto se sustenta en los conocimientos adquiridos en la etapa cualitativa, complementada por la capacidad de solución de problemas que todo ingeniero debe poseer. Esta fase ha sido orientada estrictamente al resultado, con la finalidad de realizar un sistema de control versátil, intuitivo y agradable.

Finalmente, con todos los conocimientos obtenidos y la experiencia adquirida en el desarrollo de la aplicación de control, se ha elaborado una memoria con la intención de transmitir de manera fácil y directa todo el proceso seguido. Teniendo siempre presente, el objetivo de facilitar una herramienta de conocimiento de los dispositivos y, en mayor medida, del software empleado.

5. Descripción del sistema

El sistema tiene como objetivo realizar un control de un motor mediante PAC y una interface HMI. La comunicación entre los distintos dispositivos del sistema se realiza a través del protocolo Modbus RTU y mediante el soporte físico del puerto serie RS485.

Los equipos del sistema de control son los siguientes:

- Panel de control HMI CP630
- Autómata programable AC500 PM573
- Convertidor ACS550

El sistema tiene la capacidad de controlar las acciones del motor y supervisar el estado del sistema, al mismo tiempo que permite la parametrización del control sobre el motor, tanto por interface HMI como desde el autómata programable.

La arquitectura general del sistema de control se muestra en la figura 1.

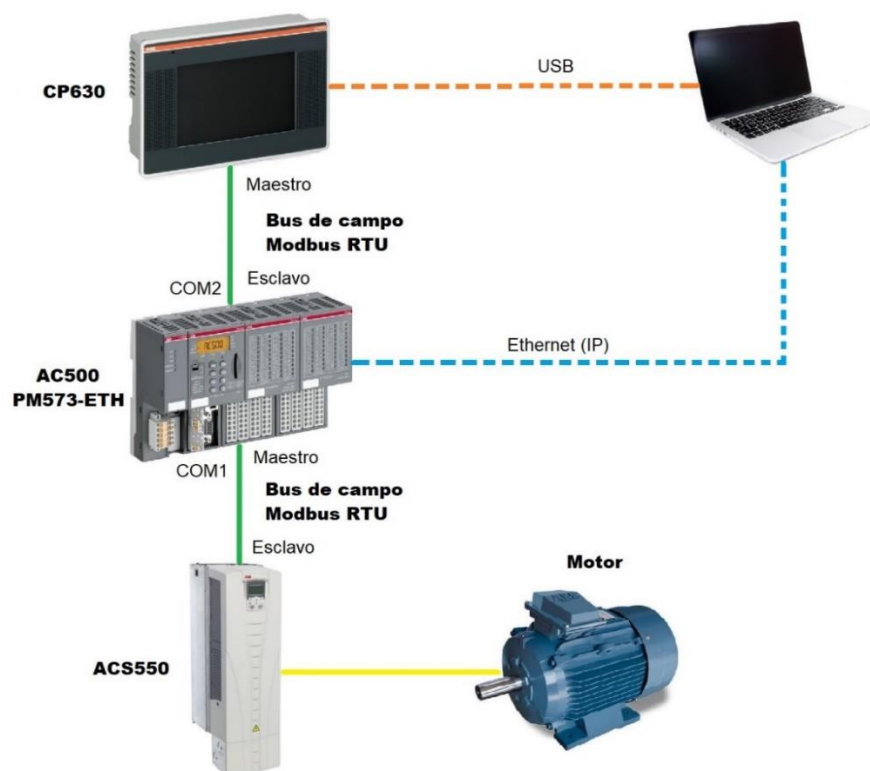


Figura 1- Arquitectura general del sistema

6. Hardware

6.1. Convertidor de frecuencia

6.1.1. Fundamentos teóricos

Los convertidores de frecuencia son dispositivos electrónicos capaces de regular la velocidad de un motor eléctrico de inducción controlando la frecuencia de entrada del motor. Se sitúan entre la red de alimentación y los bornes del motor.

Se pueden clasificar entre dos tipos de convertidores: directos e indirectos.

Los convertidores directos o cicloconvertidores están formados por un conjunto de tiristores que, realizando adecuadamente las conmutaciones necesarias, obtienen una corriente alterna de salida, resultado del sumatorio de fragmentos de onda sinusoidal precedentes de las fases de alimentación.

Son convertidores, técnicamente, complejos y de un alto coste económico. Por este motivo tienen un uso muy restringido. Su aplicación más extendida es el control de motores de alta potencia con necesidades de regulación a baja velocidad y, por tanto, de par elevado.

Los convertidores indirectos son los variadores de frecuencia más extendidos en la industria. Estos dispositivos no actúan directamente sobre la tensión de entrada, sino que tienen tres etapas básicas. En primer lugar, convierten la alimentación alterna en corriente continua a partir de un rectificador. Posteriormente, la señal se somete a un proceso de filtrado. Para finalizar en la última etapa, modulando la onda a través de un inversor, en una nueva señal alterna.

El diagrama de bloques básico del convertidor indirecto es el siguiente:

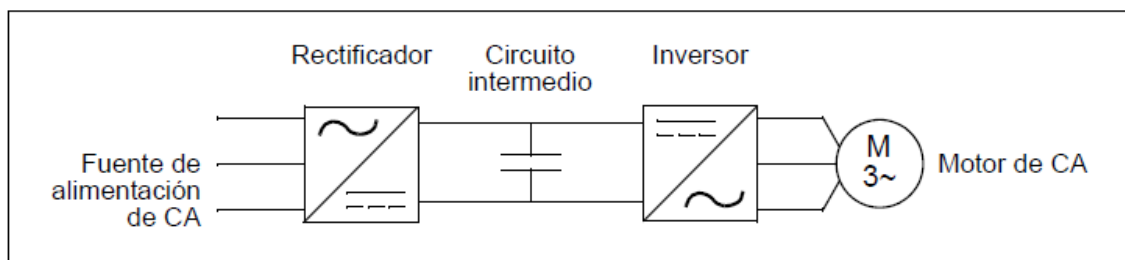


Figura 2 – Diagrama de bloques de un convertidor indirecto [30]

Como se observa en la figura 2, tenemos tres etapas: rectificador, circuito intermedio e inversor. A continuación, se analizan con más detalle.

- **Rectificador:** tiene la función de convertir la señal alterna de entrada en corriente continua y, en consecuencia, controlar la tensión que será transmitida al inversor. Se pueden diferenciar entre dos tipos de rectificadores, controlados o no controlados. Esto dependerá de los componentes y el esquema del rectificador. Los rectificadores no controlados están destinados, principalmente, a aplicaciones de baja potencia y son los más extendidos en el mercado. Los tres rectificadores más utilizados actualmente son el puente de diodos, el de tiristores y el “*DC Chopper*”.

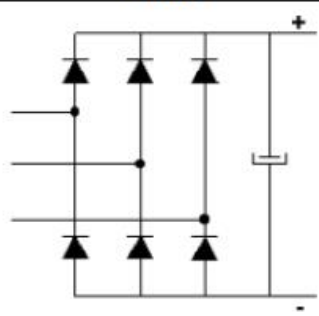
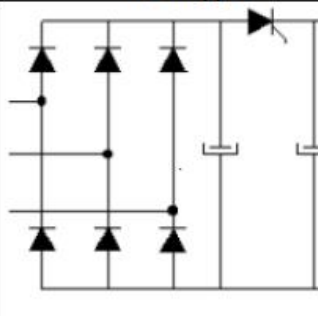
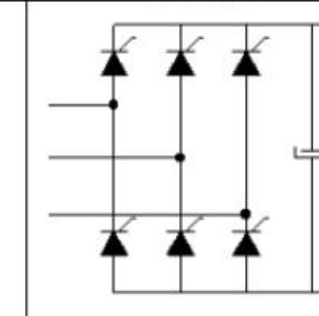
Diodos	DC Chopper	Tiristores
		
Control: No Voltaje CD: Constante Rizo CD: Bajo V/Hz: Se ajusta en Inversor Inversor: PWM FPd: Alto para todas las velocidades Armónicas: Alto Regeneración: No	Control: SCR, GTO, Transistor Voltaje CD: Variable Rizo CD: Varía V/Hz: Rectificador-Inversor Inversor: 6 pasos FPd: Se reduce con la velocidad Armónicas: Alto Regeneración: No	Control: SCR Voltaje DC: Variable Rizo CD: Varía V/Hz: Rectificador-Inversor Inversor: 6 pasos, PWM FPd: Se reduce con la velocidad Armónicas: Alto Regeneración: Sí

Figura 3 – Tipos de rectificador [32]

- **Circuito intermedio:** el objetivo de esta etapa es el filtrado de la onda, es decir, minimizar el rizado de la onda que ha sido rectificada con anterioridad. Muchos variadores tienen una salida accesible al Bus de CC, y es posible hacer uso de ella como alimentación auxiliar. En este sentido, un componente básico es el condensador electrolítico entre polaridades diferentes, con la finalidad de absorber los picos de corriente debidos a la conmutación. Sobre todo en los procesos de control del motor. Por tanto, este elemento pasivo puede ser crítico, llegando a determinar la vida útil del convertidor.

- **Inversor:** es la última etapa del convertidor. Tiene la misión de transformar la tensión del Bus CC en una tensión alterna. Esta onda final es variable en tensión y frecuencia, gracias la modulación de los pulsos. Existen onduladores de onda cuadrada, donde la onda no es estrictamente sinusoidal, pero su uso es cada vez más restringido. Los inversores más comunes en la actualidad y que ofrecen mejores prestaciones de onda son los PWM, modulación por ancho de pulsos. Estos onduladores reducen de forma drástica la presencia de armónicos, en contraposición a los onduladores de onda cuadrada. El inversor PWM está formado por seis IGBT's (Insulated Gate Bipolar Transistor) que se conmutan formando pulsos. El tiempo de conmutación varia el número y el ancho de los pulsos ajustando así la velocidad del motor. Para tiempos de ciclo mayores la velocidad será más baja, en cambio, si el tiempo de ciclo se reduce la velocidad del motor aumenta. Por tanto, con una señal de control realizamos la activación y desactivación de los IGBT's. La frecuencia de la onda final se encuentra, usualmente, entre 2 y 16 KHz. Es posible trabajar a frecuencias más altas, sin embargo, esto disminuye le rendimiento y aumenta la generación de calor. A pesar de ello, trabajar en altas frecuencias reduce drásticamente el ruido, y esto puede ser interesante en algunos entornos concretos.

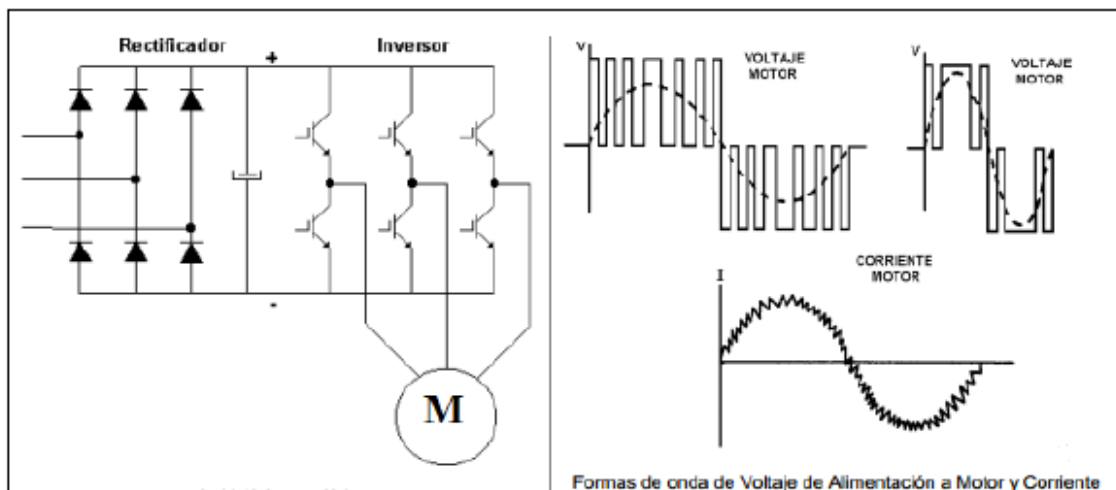


Figura 4 – Inversor PWM [32]

6.1.2. Modelo: ABB ACS550

El convertidor seleccionado es el modelo ACS550 de ABB. Un convertidor de par variable y constante, que permite el control vectorial. Dispone de un filtro EMC (Electromagnetic compatibility) y una reactancia de autoinductancia variable para reducir los armónicos. Además, viene incorporado con contadores integrados, que permiten conocer el ahorro energético, así como, las emisiones de dióxido de carbono.



Figura 5 – Vista general convertidor ACS550 [30]

El esquema con las funcionalidades e indicaciones físicas del autómata es el siguiente:

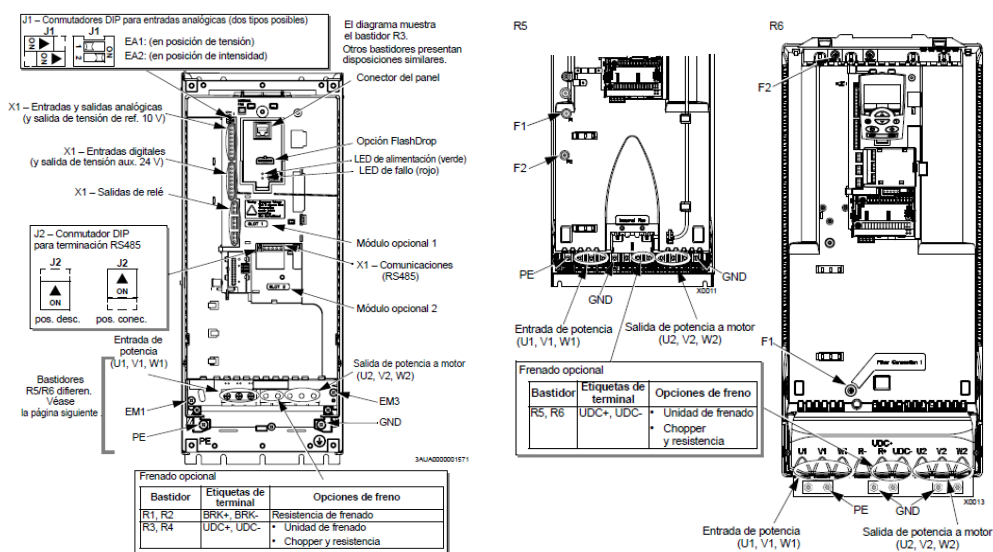


Figura 6 – Esquema de conexiones convertidor ACS550 [30]

La figura siguiente resume el funcionamiento y colocación de las teclas del panel de control asistente:

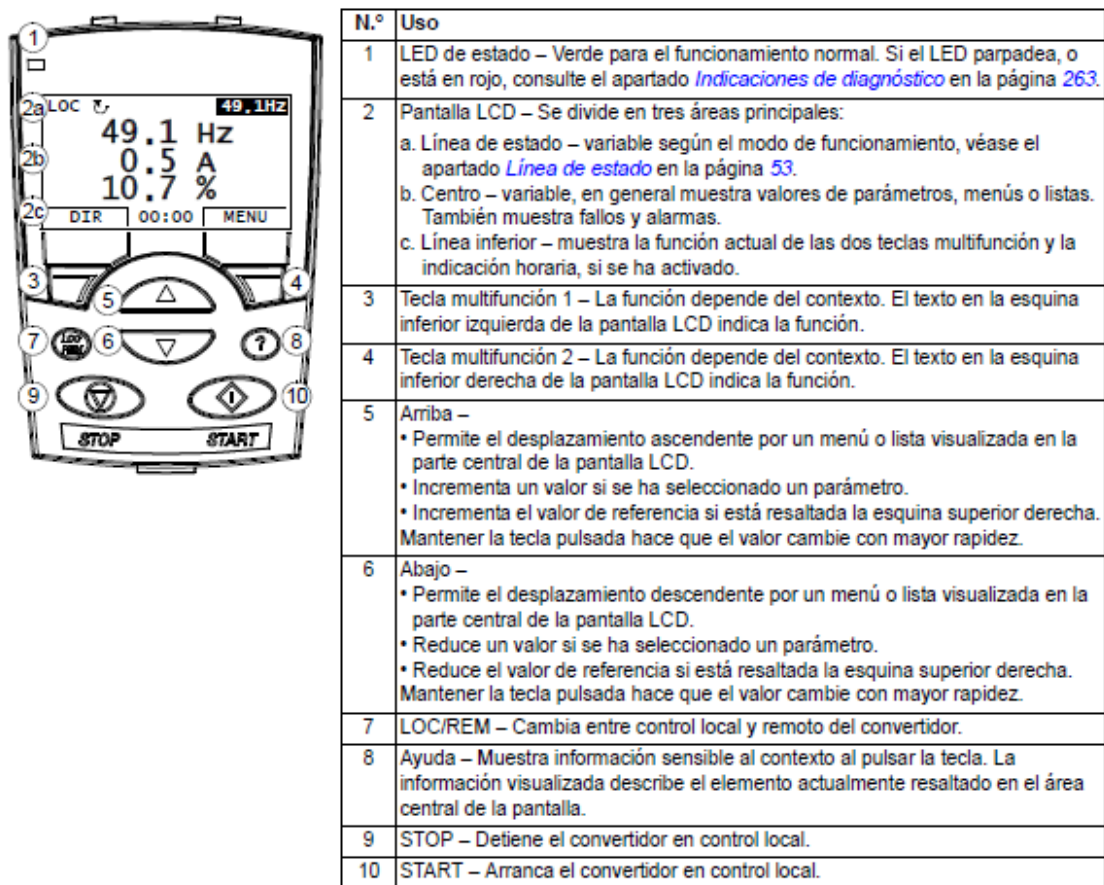


Figura 7 – Funcionalidades panel ACS550 [30]

6.1.3. Características técnicas

Los convertidores de frecuencia ACS550-01 son una gama de dispositivos con unas características y funcionalidades semejantes. Estos modelos están disponibles para diferentes potencias comprendidas entre 0,75KW y 355KW. Si nuestros requerimientos de potencia son superiores, se debería optar por un dispositivo del mismo modelo de gama superior. Todos los modelos de esta gama son estándar, es decir, tienen una interfaz común de usuario y de proceso de bus de campo, así como herramientas de software comunes.

Los convertidores ACS550 son versátiles y pueden destinarse a un amplio conjunto de finalidades. Su funcionamiento estándar hace que tengan una fácil instalación, puesta en

marcha y funcionamiento. A pesar de esto, la amplia variedad de funcionalidades y parámetros configurables permiten adaptarse a necesidades específicas y complejas.

Las características generales de los convertidores ACS550-01, independientemente del modelo seleccionado, son las siguientes:

Conexión de red	
Rango de potencia y tensión	Trifásica 380 a 480 V, +10 -15%, 0,75 a 355 KW Trifásica 208 A 240 V, +10 -15%, 0,75 a 75 KW Autoidentificación de la línea de entrada
Frecuencia	48 a 63Hz
Factor de potencia	0,98
Conexión del motor	
Tensión	Trifásica, de 0 a U_{ALIM}
Frecuencia	0 A 500 Hz
Capacidad de carga continua	Intensidad de salida nominal I_{2N} (Par constante a una temperatura máxima ambiente de 40°C)
Capacidad de sobrecarga	En uso normal $1,1x I_{2N}$ durante 1 min cada 10 min En uso de trabajo pesado $1,5x I_{hd}$ durante 1 min cada 10 min
Frecuencia de conmutación	Por defecto 4 kHz 1 kHz, 2 kHz, 4 kHz, 8 kHz, 12 kHz
Tiempo de aceleración	0,1 a 1800 s
Tiempo de desaceleración	0,1 a 1800 s
Control de velocidad	
Bucle abierto	20% del deslizamiento del motor nominal
Bucle cerrado	0,1% de la velocidad nominal del motor
Bucle abierto	+ - 5% con par nominal
Bucle cerrado	+ - 2% con par nominal
Control de par	
Bucle abierto	<10ms con par nominal
Bucle cerrado	<10ms con par nominal
Bucle abierto	+ - 5% con par nominal

Bucle cerrado	+ - 2% con par nominal
Limites ambientales	
Temperatura ambiente De -15°C a 50°C	No se permite escarcha. De 40°C a 50°C con derrateo
Altitud Intensidad de salida	Intensidad nominal disponible entre 0 y 1000m. En altitudes de 1000 a 4000m por encima del nivel del mar, el derrateo es del 1% cada 100m.
Humedad relativa	5 a 95%, sin condensación
Grado de protección	IP21 o IP54
Conexiones de control programables	
Dos entradas analógicas Señal de tensión Señal de intensidad Valor de referencia del potenciómetro Máxima demora de tiempo Resolución Precisión	0 (2) A 10V. $R_{EN} > 312 K\Omega$ sin diferencial 0 (4) A 20mA. $R_{EN} > 100 K\Omega$ sin diferencial 10V + - 2% MÁX. 10mA, $R < 10 K\Omega$ 12 a 32 ms 0,1% + - 1%
Dos salidas analógicas Precisión	0 (4) A 20mA. Carga $< 500 \Omega$ + - 3%
Tensión auxiliar	24 V DC + - 10%, máx. 250 mA
Seis entradas digitales Impedancia de entrada Máxima demora de tiempo	12 a 24 V DC con alimentación interna o externa, PNP Y NPN 2,4 K Ω 5ms + - 1ms
Tres salidas de relé Tensión máxima de conmutación Intensidad máxima de conmutación Intensidad continua máxima	250V CA / 30V DC 6A/30 V DC; 1.500V A/230 V AC 2 A rms
Comunicación serie EIA-485	Protocolo MODBUS
Control y opciones de comunicación	
Fieldbusses	DeviceNet, PROFIBUS, CANopen, ControlNet,

	EthernetIP, EtherCAT, Modbus TCP, PROFINET, PowerLink
Características adicionales	
Contadores de eficiencia energética, Analizador de carga, Herramienta FlashDrop, Asistentes de puesta en marcha, Optimización del ruido, Filtro EMC integrado, Chopper de frenado, Reactancia de autoinductancia patentada	
Cumplimiento normativas de producto	
Directiva Europea de Baja tensión 2006/95/CE Directiva europea sobre maquinas 2006/42/CE Directiva EMC 2004/108/CE Directiva RoHS	

Tabla 1 – Características técnicas convertidor ACS550

6.1.4. Macros y configuración de parámetros

Los variadores de frecuencia ACS550 disponen de perfiles con parámetros prediseñados llamadas macro. Estas macros modifican un grupo de parámetros en unas variables concretas. Esto permite una configuración rápida de sistemas con una finalidad específica.

Cuando se modifica una macro, no solamente se seleccionan unos parámetros concretos, sino que los demás vuelven a sus valores por defecto. Por este motivo, se deben revisar la lista de parámetros, si algunos parámetros han sido modificados por el usuario con anterioridad. Existen módulos relativos a la seguridad del sistema que si han sido modificados no serán reemplazados.

Una vez se ha seleccionado una macro, el usuario puede modificar parámetros, y por tanto, seleccionar el perfil que más se asemeje a sus necesidades. Modificando, únicamente, las especificaciones concretas que necesite. Cada macro está destinada a una finalidad y un tipo de control concreto, por ello, es posible que el conexionado del variador varíe, y por tanto, debe consultarse el manual para cablear correctamente las entradas y salidas del variador.

Las macros de aplicación se habilitan ajustando el valor de parámetro “9902”. Por defecto, tendremos seleccionada la macro 1 (Perfil ABB). Las macros que podemos seleccionar en los modelos ACS550 son:

Macro	Características
Estándar ABB	Macro por defecto. Proporciona una configuración de E/S de 2 hilos de cometido general con 3 velocidades constantes. Los valores de parámetros son los predeterminados por defecto.
3 Hilos	Se utiliza cuando hacemos un control mediante pulsadores. Proporciona tres velocidades constantes. Tienen una entrada de inhabilitación del marcha/paro.
Alterna	Ofrece una configuración de E/S adaptada para una control del motor con cambio de sentido de rotación.
Potenciómetro del motor	Se emplea para variar la velocidad a partir de PLC variando únicamente señales digitales.
Manual-Auto	Esta macro proporciona una configuración de E/S que normalmente se utiliza en aplicaciones HVAC.
Control PID	Proporciona ajustes de parámetros para sistemas de control en bucle cerrado como el control de presión o flujo.
PFC	Los ajustes de parámetros están destinados a aplicaciones de control de bombas y ventiladores (PFC)
Control de par	Proporciona ajustes de parámetros para aplicaciones que requieren el control de par. Este ajuste puede conmutarse para que control de velocidad.

Tabla 2 – Macros de aplicación ACS550

La macro seleccionada en el presente proyecto es la número 1, Estándar ABB. Es posible ver que macro ha sido seleccionado tanto en el programa del autómatas como en la pantalla HMI. El valor de la macro no es editable en el sistema, ya que, al modificarla podría reemplazar parámetros necesarios para el control por bus de campo.






Los variadores ACS550, además de seleccionar perfiles, permiten la modificación de parámetros para realizar una configuración ajustada al proceso. Los parámetros están agrupados por familia según la finalidad de cada uno de ellos. Se pueden modificar por bus de campo, o bien, desde la pantalla de control del mismo variador. Cada parámetro se define por una dirección de cuatro números, donde los dos primeros corresponden al grupo de la familia que pertenecen.

Los grupos de parámetros existentes son las siguientes:

Código	Grupo
99	Datos de partida
01	Datos de funcionamiento
03	Señales ACT bus de campo
04	Historial de fallos
10	Marcha/Paro/Dirección
11	Selección referencias
12	Velocidades constantes
13	Entradas analógicas
14	Salidas de relé
15	Salidas analógicas
16	Controles sistema
20	Limites
21	Marcha / Paro
22	Aceleración / Deceleración
23	Control velocidad
24	Control par
25	Velocidades criticas
26	Control motor
29	Mantenimiento
30	Funciones fallos
31	Rearme automático
32	Supervisión
33	Información
34	Pantalla panel
35	Temporizador motor
35	Funciones temporizadas
37	Curva de carga
40	CONJ PID proceso 1
41	CONJ PID proceso 2
42	PID TRIM/EXT

45	Ahorro energético
50	Encoder
51	Mod comunicación ext
52	Comunicación panel
53	Protocolo BCI
64	Analizador de carga
81	Control PFC
98	Opciones

Tabla 3 – Grupo de parámetros ACS550

Para modificar los parámetros desde el convertidor se debe seleccionar la tecla menú , mediante las teclas de posición ( y ) nos dirigimos a “PARAMETROS” y pulsamos . Nos aparecerá la lista de grupos de parámetros. A continuación, seleccionamos la familia que se quiera modificar y pulsamos INTRO. Seguidamente, encontraremos la lista de parámetros de la familia seleccionada, nos movemos con las teclas de posición hasta encontrar el parámetro a modificar y pulsamos INTRO. Aquí podremos modificar el valor del parámetro. En algunos casos, los valores son numéricos y representativos de una función, el significado de cada valor debe consultarse en el manual. Finalmente, para salir pulsamos .

Si se han realizado cambios de parámetros, es posible guardar la configuración como si fuese una macro. El sistema permite guardar dos perfiles adicionales. Para ello, se debe seleccionar el parámetro 9902. Esta opción tiene la finalidad de guardar perfiles, en caso de realizar una aplicación con finalidades diversas y si se debe modificar con frecuencia. Si se desea guardar la configuración para poder modificarla, se debe realizar una copia de seguridad dirigir en el menú del convertidor a “Salvar parámetros”.


6.1.5. Sistemáticas de funcionamiento



El control del motor a través del variador de frecuencia se puede realizar de diversas maneras. En cualquier caso, antes de realizar el sistema de control es necesario caracterizar el motor.

Para caracterizar el motor en la puesta en marcha inicial, se deben indicar los parámetros nominales del motor en el grupo 99 (DATOS DE PARTIDA). Una vez introducidos estos valores, es necesario realizar una “Marcha ID”. Con este sistema, el convertidor calcula, automáticamente, las características del motor mediante magnetización de identificación. Es un proceso de autocalibración, durante este proceso el convertidor acciona el motor hasta un 50 o un 80 por ciento de la potencia nominal. Por este motivo, se debe estar seguro de liberar el motor de la carga. Una vez finalizada la marcha, el convertidor crea un modelo a partir de cálculos internos que permitirá un control eficaz, sobre todo, si se realiza un control de velocidad o de par.

La modificación de algunos parámetros característicos del motor puede requerir realizar una nueva “Marcha ID”, en este caso, el propio variador nos lo indicará en el panel y bloqueará el accionamiento del mismo.

Una vez realizada la puesta en marcha inicial del variador con su correspondiente caracterización del motor, se puede emprender la labor de configurar el sistema de control.

El convertidor se puede controlar de manera local o de forma remota, para ello, únicamente se debe seleccionar la tecla del panel de control .

El modo local es el control a través del panel de control del propio variador. Para realizar este sistema, solamente, necesitamos configurar los parámetros que nuestra aplicación necesite, o bien, seleccionar una macro. A continuación, para para arrancar el convertidor pulsamos  y para detenerlo .

En modo remoto, se puede controlar el variador desde dos sistemas distintos. A partir de las propias entradas y salidas del convertidor, o en su defecto, por el bus de campo. El control por bus de campo, concretamente MODBUS RTU, se tratará con más profundidad en los capítulos posteriores, desde un sistema basado en un autómata programable y desde un sistema HMI.

El control en modo remoto desde las propias E/S del variador, puede admitir desde un sistema básico hasta uno de complejidad alta. Aun así, las bases fundamentales son semejantes. Sin entrar en detalle del control del motor, para realizar la configuración de un sistema por control de E/S con cambio del sentido de giro realizaremos el siguiente setup en las conexiones del variador:

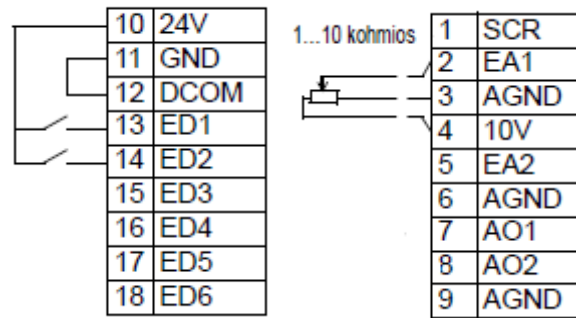


Figura 8 – Control ACS550 mediante E/S del convertidor [30]

Una vez se ha cableado las entradas del convertidor, se selecciona la macro de aplicación “Estándar ABB”, para ello como se ha visto en el capítulo anterior, se debe dirigir al parámetro “9902”. A continuación, para poder realizar la inversión de giro se debe modificar el parámetro “1003” al valor “PETICION”, que se identifica en el variador con el número 3. Finalmente, comprobamos que el convertidor se encuentra en modo remoto.

Con esta sencilla configuración, con la activación/desactivación de la entrada ED1 realizamos el marcha/paro del motor. La entrada ED2, nos permite seleccionar el sentido de giro del motor. Actuando sobre el reóstato, variamos la tensión de la entrada analógica EA1, esto nos permitirá regular la frecuencia del motor y, por tanto, su velocidad.

6.2.PAC

6.2.1. Fundamentos teóricos

Los autómatas programables son dispositivos que tienen el objetivo de almacenar y ejecutar los programas de control, mediante el cual, es posible monitorizar procesos en tiempo real y realizar aplicaciones secuenciales.

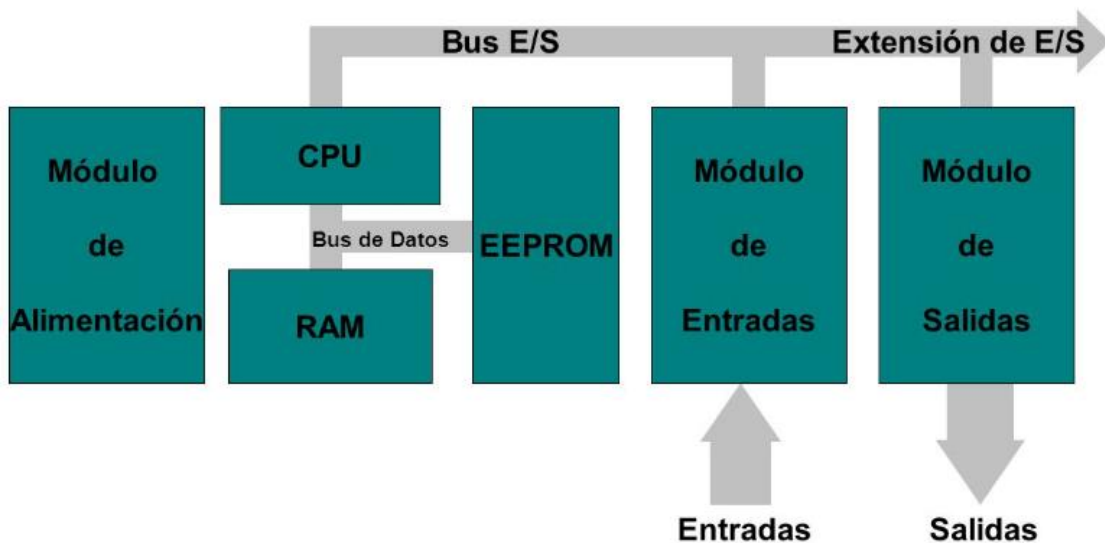


Figura 9 – Diagrama básico de un PAC.

Curso capacitación PLC. Recuperado de: <https://slideplayer.es/slide/4307257/> (2019)

Los autómatas programables recibían el nombre de PLC's (Programmable Logic Controller) porque en sus inicios procesaban exclusivamente señales digitales y, en consecuencia, únicamente realizaban operaciones lógicas básicas. Además, eran dispositivos compactos que difícilmente era posible modificar su configuración.

En la actualidad, los autómatas programables tienen la capacidad de trabajar con señales analógicas, PID's, comunicaciones, o incluso, funciones de matemática avanzada. Asimismo, no solo han sufrido cambios internos en sus capacidades, sino que los modelos que podemos encontrar en el mercado actual, suelen ser modulares, lo cual, permite adaptar un dispositivo a las necesidades de un proyecto, o en su defecto, destinarlo a usos concretos muy diferenciados entre sí.

Según la gama de un autómata, podemos encontrar diferentes funcionalidades o capacidades. Sin embargo, independientemente del fabricante y la gama del modelo, los bloques básicos que componen un autómata son:

- **CPU:** Unidad central de proceso. Es el núcleo de la arquitectura del autómata. Su misión es el procesamiento del programa, y por tanto, supervisar y controlar el sistema. Está basado en un microcontrolador que determinará las características del autómata.
- **Memoria interna:** Existen tres tipos de memoria. La memoria del programa, destinada a guardar el código de procesamiento. La memoria de imagen, destinada a guardar los estados de las E/S, así como variables de memoria interna. Y finalmente, la memoria EPROM que es la encargada de guardar el firmware y los archivos temporales.
- **Módulos E/S:** Es el módulo que interactúa con el proceso, encargado de dar y recibir las órdenes del sistema. Existen de tres tipos: digitales, analógicos o especiales (contadores, comunicaciones, especiales para motores, etc...). Debido a la variedad de propósitos, estos módulos pueden trabajar a diferentes tensiones.
- **Fuente de alimentación:** es la parte encargada de transformar la energía eléctrica de red a la tensión de trabajo de los dispositivos. Puede tener el objetivo de dar la potencia necesaria al autómata, o bien, proveer también los dispositivos y módulos que se dispongan en las entradas y salidas.

Los autómatas programables son un elemento clave en la automatización de los procesos, y en muchos casos también, un elemento fundamental de la robótica básica. En este sentido, la utilización de autómatas nos permite una facilidad de operación y monitorización con una relativa facilidad de instalación, seguridad en los procesos y la posibilidad de cambios sin necesidad de rediseñar una instalación. Si añadimos su escaso mantenimiento, puede proporcionar una disminución de costes considerable. Aunque se requiera de una inversión inicial elevada, los nuevos procesos de estandarización y su implantación generalizada están reduciendo costes y, por tanto, hacen su utilización cada vez más atractiva y necesaria para enfrentarse a los nuevos retos de un mercado globalizado.

6.2.2. Autómatas programables: ABB

Los autómatas programables de ABB, conocidos como la gama AC500, se lanzaron al mercado en 2006. Disponen de una amplia gama de componentes escalables, flexibles, de alto rendimiento, calidad y fiabilidad.



Figura 10 - Autómata escalable AC500 [27]

La gama de PAC's AC500 se caracteriza por integrar canales digitales configurables, como entradas o salidas, una gran variedad de comunicaciones, llegando incluso a funcionar a través de diferentes buses de campo. La flexibilidad escalable permite múltiples combinaciones que se ajusten a las necesidades de cada proyecto.

Dentro de la gama AC500 de autómatas de ABB, se pueden distinguir los siguientes modelos según las características del proyecto:

- **AC500:** Es el PAC de referencia y presenta una gran variedad de prestaciones, comunicaciones y capacidades de E/S para aplicaciones industriales. La opción ideal para soluciones de maquinaria de alta velocidad y de redes complejas entre otras.
- **AC500-eCo:** Este PAC compacto ofrece flexibilidad y configuraciones competitivas para su sistema de control. La mejor opción soluciones de automatización en aplicaciones sencillas.
- **AC500-S:** Este PAC de seguridad ha sido diseñado para aplicaciones de seguridad en áreas de automatización como maquinaria o factorías. La solución ideal para aplicaciones de seguridad desde las más sencillas hasta las más complejas.

- **AC500-XC:** Es el PAC idóneo para aplicaciones en condiciones extremas y que pueden requerir de un rango de temperatura extendido, altura elevada, inmunidad a vibraciones, gases peligrosos y entornos salinos, entre otros.

Los autómatas AC500 cumplen con la normativa de seguridad eléctrica EN 61131-2, además de la normativa aplicable sobre compatibilidad electromagnética EN 61000. Asimismo, toda la familia de productos AC500 tienen una clasificación IP67, lo cual, demuestra la robustez de los componentes del sistema y permite su instalación en entornos con grados de polución alta.

Las posibilidades de intercambio de los modelos AC500 residen en su montaje en una base de terminales. El tipo de base depende del número de módulos y las comunicaciones que requiere el proyecto. Las bases de terminales están diseñadas para permitir una rápida conexión y desconexión sin necesidad de instalaciones, cableado o herramientas.

Las CPU'S son sustituibles y tienen el mismo formato. Normalmente, están diseñadas con microcontroladores de 8, 16 o 32 bits. Se componen de una ranura SD, FBP, 2 puertos serie, pantalla, teclado y cavidad para pila de litio. Pueden soportar un máximo de 10 módulos de ampliación.

Los módulos de entradas y salidas disponibles pueden ser tanto digitales como analógicos. Las entradas digitales son configurables como entradas o salidas y, por tanto, no tienen una naturaleza constructiva concreta. Los canales analógicos se caracterizan por ser multirango. Asimismo, los módulos de E/S se pueden ubicar en la base de terminales, a partir de una fácil instalación, o bien, remotamente mediante unas cabeceras específicas. Las tensiones de trabajo pueden ser 24V DC o 230V AC.

6.2.3. Terminal base ABB TB511

La base de terminales para conectar la CPU y los módulos extensibles es el modelo TB511-ETH. Dispone de la posibilidad de montar un módulo de comunicaciones. Las conexiones disponibles son las siguientes:

- Alimentación: 24 V DC
- COM1: bloque de terminales
- COM2: 9 polos SUB-D
- Ethernet: RJ45
- FielBusPlug



Figura 11 – Vista general Base de Terminales TB511 [19]

6.2.4. PAC ABB PM573 – ETH

El autómata seleccionado para la realización del presente proyecto es el modelo PM573-ETH de ABB. Es un autómata compacto y modular con tiempos de ciclo bajos. Con posibilidad de conexión de hasta 10 módulos. Los puertos series permiten la conexión remota de los módulos. Dispone de una interfaz Ethernet integrada que permite su uso para programación o comunicación, es decir, como servidor web o FTP.



Figura 12 – Vista general autómata PM573-ETH [29]

El esquema con las funcionalidades e indicaciones físicas del autómata es el siguiente:

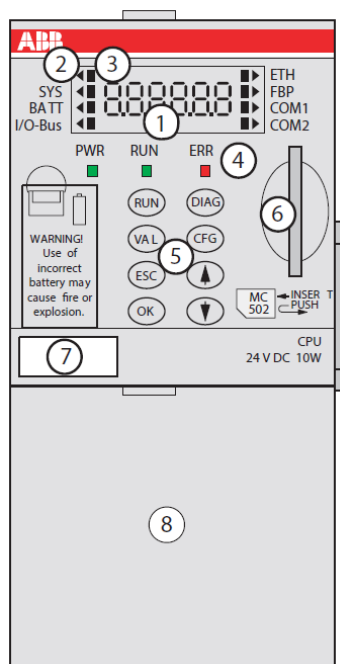


Figura 13 – Funcionalidades panel PM573-ETH [29]

- 1.-Indicaciones de estado
- 2.-Indicaciones triangulares para “Ítem”
- 3.-Indicaciones cuadradas para “Estado”
- 4.-LEDs de estado
- 5.-Teclas de función
- 6.-Ranura tarjeta de memoria SD
- 7.-Etiqueta
- 8.-Conexiones de la base de terminales

Las características técnicas del modelo PM573-ETH seleccionado se detallan a continuación:

Característica	PM573-ETH
Tensión nominal	24V DC
Potencia	10 W
Memoria Flash EPROM and RAM	512 kB
Tarjeta de memoria	Mínimo 512 kB
Datos del servidor WEB	1024 kB
Tiempo de ciclo por instrucción	
Binary	0,07 μ s
Word	0,09 μ s
Floating-point	0,70 μ s
Número máximo E/S	
Entradas digitales	320
Salidas Digitales	240
Entradas analógicas	160
Salidas analógicas	160
Ejecución del programa	
Cíclico, Tiempo controlado, Tiempo real de reloj, Multi-tarea	
Conexiones	
COM1	RS232/RS485 configurable. Bloque de terminales
COM2	RS232/RS485 configurable. SUB-D hembra, 9 polos.
FieldBusPlug	M12 macho, 5 polos. PROFIBUS, CANopen, DeviceNet)
Ethernet	RJ45

Tabla 4 – Características técnicas autómatas PM573-ETH

6.2.5. ABB DA501

El módulo de expansión DA501 es un módulo mixto de entradas y salidas digitales y analógicas. Todas las entradas tienen una resolución configurable de -10...+10 V, 12 bits + signo, 0...10 V, 0...20mA, 4...20mA. Se puede realizar la conexión mediante el bus lateral de terminales o con una longitud máxima de cableado de hasta 1000 metros para un procesamiento de señales correcto. Se recomienda realizar el conexionado con par trenzado, o en su defecto, cable apantallado.



Figura 14 – Vista general módulo DA501 [22]

El esquema con las funcionalidades e indicaciones físicas del módulo DA501 es el siguiente:

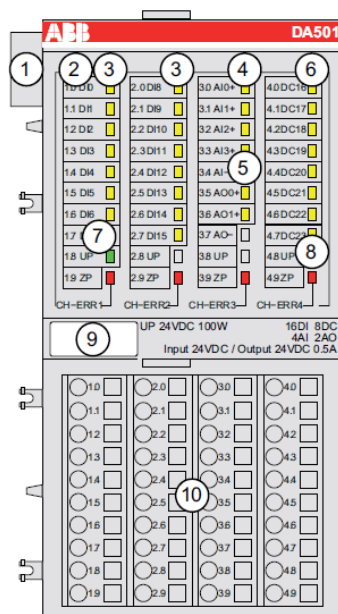


Figura 15 - Funcionalidades módulo DA501 [22]

- 1.- Bus E/S
- 2.- Asignación número de borne – nombre de señal
- 3.- LEDs amarillos indicadores de señal (DI0 – DI15)
- 4.- LEDs amarillos indicadores de señal (AI0 – AI13)
- 5.- LEDs amarillos indicadores de señal (AO0 – AO1)
- 6.- LEDs amarillos indicadores de señal (DC16 – DC23)
- 7.- LED verde para tensión de alimentación
- 8.- LEDs rojos indicadores de error
- 9.- Etiqueta
- 10.- Bloque de bornes

Las características técnicas del modelo DA501 seleccionado se detallan a continuación:

Parámetro	Valor
Entradas digitales	16 (24V DC, tipo 1, retardo configurable)
E/S digitales configurables	8 (24 V DC, 0,5 A máx.)
Entradas analógicas	4 (Resolución: 12 bits + signo, tensión o corriente)
Salidas analógicas	2 (Resolución: 12 bits + signo, tensión o corriente)
Alimentación principal	A partir de la tensión de alimentación del proceso UP
Alimentación interna	A través de la interfaz del bus de expansión (bus de E / S)
Alimentación externa	Vía terminales UP y ZP (suministro 24 VDC)
Efecto de la conexión incorrecta del terminal de entrada	Señal errónea o no detectada, ningún daño hasta 35 V

Tabla 5 – Características técnicas módulo DA501

6.2.6. ABB CD522

El módulo de expansión CD522 es un módulo para encoder y señales PWM. El módulo dispone de 2 entradas de encoder, 2 salidas PWM, 2 entradas digitales y 8 canales configurables como entrada o salida. Los encoders trabajan de forma independiente uno del otro y son configurables por software mediante 10 modos de operación para frecuencias hasta 300 kHz. Se puede realizar la conexión mediante el bus lateral de terminales o con una longitud máxima de cableado de hasta 1000 metros para un procesamiento de señales correcto. Se recomienda realizar el conexionado con par trenzado, o en su defecto, cable apantallado.



Figura 16 – Vista general del módulo CD522 [22]

El esquema con las funcionalidades e indicaciones físicas del módulo CD522 es el siguiente:

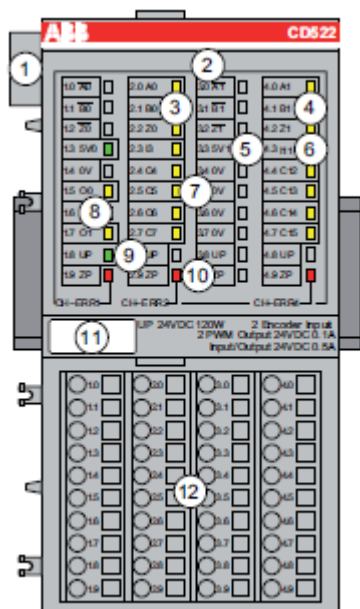


Figura 17 - Funcionalidades módulo CD522 [22]

- 1.- Bus E/S
- 2.- Asignación número de borne – nombre de señal
- 3.- LEDs amarillos indicadores estado de señal encoder 0
- 4.- LEDs amarillos indicadores estado de señal encoder 1
- 5.- LEDs verdes de estado alimentación 5V
- 6.- LEDs amarillos indicadores estado de señal
- 7.- LEDs amarillos indicadores estado de señal entrada/salida
- 8.- LEDs amarillos indicadores estado de señal PWM/Pulse
- 9.- LED verde para tensión de alimentación
- 10.- LEDs rojos indicadores de error
- 11.- Etiqueta
- 12.- Bloque de bornes

Las características técnicas del modelo DC522 se detallan a continuación:

Parámetro	Valor
Entradas/Salidas digitales	<p>24 VDC, las entradas / salidas dedicadas se pueden usar para funciones de conteo:</p> <ul style="list-style-type: none"> - Operación de captura / toque, valor del contador almacenado en variable separada en evento externo (flanco ascendente o descendente) - Ajuste la entrada al registro de contador preestablecido con un valor predefinido - Establecer entrada para restablecer el registro del contador - Salida de valor final; la salida se establece cuando el valor predefinido es alcanzado - Entrada de inicialización del punto de referencia (RPI) para incremental. <p>Todas las entradas / salidas no utilizadas pueden utilizarse con la especificación de</p>

	rango de entrada / salida estándar. Efecto de la conexión incorrecta del terminal de entrada: incorrecto o no señal detectada, sin daños hasta 35 V.
Contador /Encoder	Integrado, 2 contadores (interfaz de hardware con +24 VDC, +5 VDC, diferencial y entrada sinusoidal de 1 Vpp) con hasta 12 configurables modos de operación: <ul style="list-style-type: none"> - 32 bits un modo de contador - Modo de contador de 16 bits dos. - Encoder incremental de posición. - Absolute SSI encoder - Medidor de frecuencia de tiempo - Entrada de frecuencia hasta 300 kHz.
PWM/Pulse salida	2 moduladores de ancho de pulso o salidas de pulso Especificación de salida: <ul style="list-style-type: none"> - Salida push-pull: 24 VDC, 100 mA máx. - Limitación de corriente (térmica y sobrecorriente) Especificación de PWM <ul style="list-style-type: none"> - Frecuencia de 1 Hz a 100 kHz. - Valor del 0 al 100%. Especificación de pulso <ul style="list-style-type: none"> - Frecuencia de 1 Hz a 15 kHz. - Emisión de pulsos de 1 a 65535 pulsos. - Indicador de número de impulsos emitidos (0 a 100%) Especificación de frecuencia <ul style="list-style-type: none"> - Salida de frecuencia = 100 kHz cuando el ciclo de trabajo se establece en 50%
Alimentación encoders	5 V, máx 100mA
Alimentación interna	A través de la interfaz del bus de expansión (bus de E / S)
Alimentación externa	Vía terminales UP y ZP (suministro 24 VDC)

Tabla 6 - Características técnicas módulo CD522

6.2.7. Configuración física del autómata

La configuración del autómata quedaría compuesta por los componentes anteriormente detallados y su emplazamiento físico se puede observar en el siguiente diagrama de bloques:

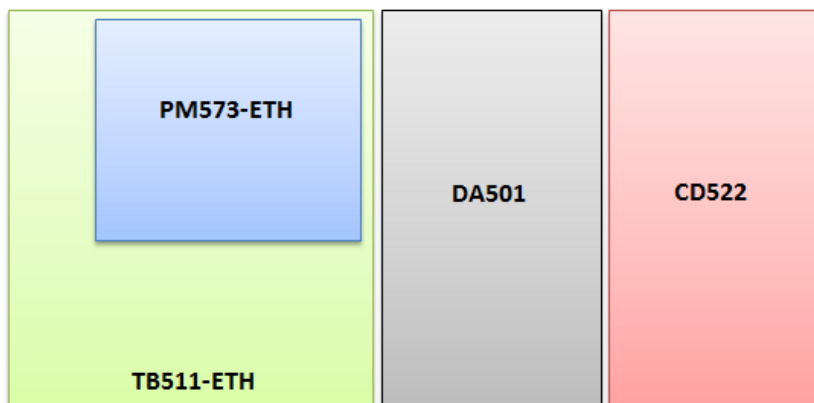


Figura 18 – Configuración del autómata

6.3.HMI

6.3.1. Fundamentos teóricos

Un sistema HMI (interface Hombre-Máquina) es el punto de encuentro entre el sistema y el operario. No tiene por qué ser un sistema complejo. Un simple accionamiento marcha/paro puede suponer el único punto de encuentro entre una máquina y el operario.

Sin embargo, los sistemas de control han avanzado considerablemente en los últimos años. Desde sistemas supervisados por un operario de manera constante en paneles con aparatos de medida analógicos, hasta la actualidad, donde en un soporte móvil tenemos la capacidad de tener los datos y las incidencias de un sistema en tiempo real.

Un buen sistema HMI debe tener unas características básicas:

- **Eficacia:** el sistema debe realizar correctamente la tarea que el operario desea ejecutar.
- **Eficiencia:** las tareas deben ser intuitivas y poder desarrollarlas con facilidad sin posibilidad de error.
- **Satisfacción:** Debe mejorar el manejo del sistema y realizar las tareas con mayor comodidad.

Uno de los sistemas HMI más extendidos en la industria es el SCADA (Supervisory Control And Data Adquisition). Es un software particularmente gráfico que permite la adquisición de datos, la supervisión del sistema y el control del proceso. Asimismo, debe ser un sistema flexible que permita la evolución de procesos industriales en constante cambio, al mismo tiempo, que proporciona una interface fácil de usar.

El SCADA no realiza directamente las funciones de control, sino que se comunica con los diferentes dispositivos del sistema que finalmente ejecutan las ordenes. En el caso del presente proyecto, el sistema HMI indica las ordenes al autómatas programable que, a su vez, las ejecutará el variador de frecuencia.

En la actualidad, los sistemas HMI deben ofrecer la posibilidad de gestionar alarmas y datos, a la vez que posean la capacidad de guardar las incidencias para analizarse con posterioridad. Teniendo la capacidad de enviar la comunicación a sistemas integrados de automatización, o sistemas integrados ERP.

Algunos de los beneficios que puede aportar la implementación un sistema SCADA son:

- Reducción de costes operacionales
- Simplificación en la interacción entre distintos sistemas
- Aumento de la calidad por la disminución de errores evitables
- Rápida respuesta ante imprevistos
- Monitorización de datos a distancia
- Análisis del sistema para una mejora continua
- Mejora del mantenimiento preventivo
- Simplificación y facilidad de uso en el control de los procesos

6.3.2. Modelo: ABB CP630

El panel de control CP630 es un terminal táctil de 5,7 pulgadas. Tiene un recubrimiento robusto de aluminio. Es un equipo de alto rendimiento, destinado a la automatización de fábricas y el control de máquinas.

Está diseñado para ejecutar el software de Panel Builder 600 (incluida la última versión PB610). Entre sus especificaciones más importantes se encuentra tener un soporte grafico de vector completo que permite la visualización de formato numérico, texto, gráfico de barras e indicadores analógicos. Además, es compatible con las funciones HMI de última generación como la adquisición de datos, manejo de alarmas, correo electrónico y menús rotativos. Todas estas funcionalidades en soporte multilinguaje.



Figura 19 – Vista general del panel CP630 [15]

El dispositivo CP630 es táctil, y por tanto, dispone de una superficie limpia sin botones. La única indicación del panel es un LED de encendido. El diagrama con las conexiones existentes en el modelo CP30 es el siguiente:

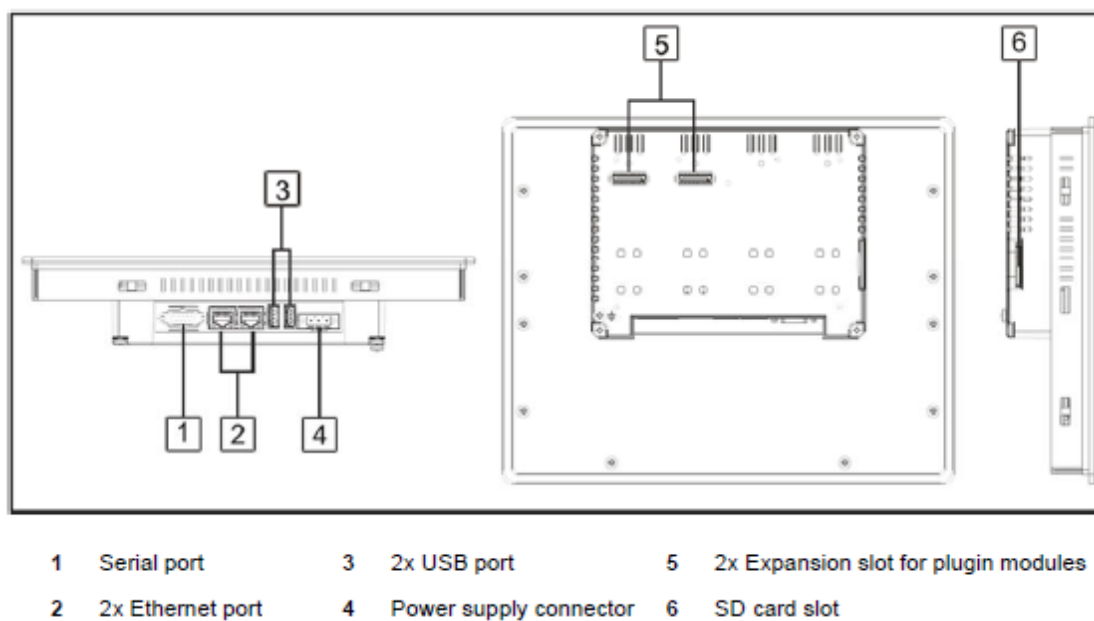


Figura 20 – Conexiones del panel CP630 [15]

6.3.3. Características técnicas

Los dispositivos CP600, y en concreto el modelo CP630, son terminales de alto rendimiento. En su parte frontal tiene una clase IP66, ideal para entornos industriales. Es un

dispositivo de bajo consumo eléctrico y dispone de una batería de litio recargable como backup.

Las principales carteristas técnicas del panel de control CP630 se detallan a continuación:

Parámetro	Valor
Pantalla	
Tamaño	5,7"
Resolución	320x240 pixels
Tipo	TFT-LCD
Numero de colores	65536
Brillo	200 cd/m ²
Iluminación	LED (40 000 h at 25°C)
Táctil	
Tipo	Resistiva de un solo toque, 4 cables
Material	Cristal recubierto de película de plástico
Recursos del sistema	
Procesador	ARM Cortex-A8, 600 MHz
Sistema operativo	Microsoft Windows 6.0 Core
Memoria RAM	256 MB
Memoria FLASH	128 MB
Ranura para tarjetas	SD
Batería	
Tipo	3 V, 50 mAH, Litio, no reemplazable y recargable
Reloj a tiempo real	
Exactitud	< 100 ppm at 25 °C
Parámetros eléctricos	
Tensión	24 V DC
Corriente	0,7 A
Protección polaridad inversa	Si

Condiciones ambientales	
Temperatura	Operación: 0...50°C Almacenaje: -20...+70°C Transporte: -20...+70°C
Humedad	5...85 % humedad relativo sin condensación
Conexiones	
USB, puerto serie (RS232, RS485, RS422), Ethernet	

Tabla 7 - Características técnicas panel CP630

6.4.Instrucciones de conexionado

6.4.1. Conexión PC – PAC

El conexionado entre el PC y el automata programable se realiza a través del protocolo TCP(Transmission Control Protocol). Es un sistema que comprende un conjunto de protocolos que cubren diferentes niveles del modelo OSI.

El protocolo TCP se aplica a través de cable Ethernet mediante IP, protocolo de internet. El sistema IP establece direcciones que identifican a cada dispositivo para el intercambio de datos(Para ver la configuración IP de los dispositivos vease apartado 7.2.5). La dirección IP empleada para la transmisión de datos es v4, que se caracteriza por tener 32 bits.

La comunicación se realiza a través de cable Ethernet cruzado, que permite la conexión de los dos dispositivos en modo full duplex.

6.4.2. Conexión PAC – Variador de frecuencia

La conexión entre el autómata programable y el convertidor se realiza a través de puerto serie con el protocolo RS485, también conocido como EIA-485. Este modo de transmisión se caracteriza por ser un bus lineal multipunto con terminadores de línea, que impiden que la señal rebote. Puede transmitir a velocidades de entre 9,6 kbps y 12 Mbps. Permite la conexión de 32 equipos (estaciones), llegando a poder conectarse 127 con la utilización de repetidores. La comunicación se realiza en half dúplex y con una distancia máxima de 1200 metros.

La conexión se realiza mediante un cable de par trenzado con apantallamiento de cobre. Los dos dispositivos disponen de bloques de terminales para realizar la conexión. En las figuras 22 y 23, podemos observar la correspondencia de cada uno de sus pins para realizar la conexión.

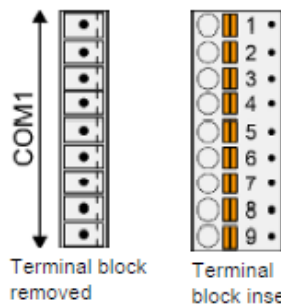
		Pole	Signal	Interface	Description
	1	1	Terminator P	RS-485	Terminator P
	2	2	RxD/TxD-P	RS-485	Receive/Transmit, positive
	3	3	RxD/TxD-N	RS-485	Receive/Transmit, negative
	4	4	Terminator N	RS-485	Terminator N
	5	5	RTS	RS-232	Request to send (output)
	6	6	TxD	RS-232	Transmit data (output)
	7	7	SGND	Signal Ground	
	8	8	RxD	RS-232	Receive data (input)
	9	9	CTS	RS-232	Clear to send (input)

Figura 21 – Pins de conexionado COM1 PM573-ETH [28]

En la figura 22, se observa la topología de la conexión realizada entre el autómatas y el convertidor. El bloque de terminales del autómatas se representa por los cuatro primeros pins, puesto que son los correspondientes al protocolo RS485. En el variador se interconecta una resistencia de 120Ω , entre los polos de transmisión, para indicar el final de línea. En cambio, en el autómatas la resistencia de 120Ω , se conecta entre polos de transmisión y los polos P-N, para indicar además su función de maestro.

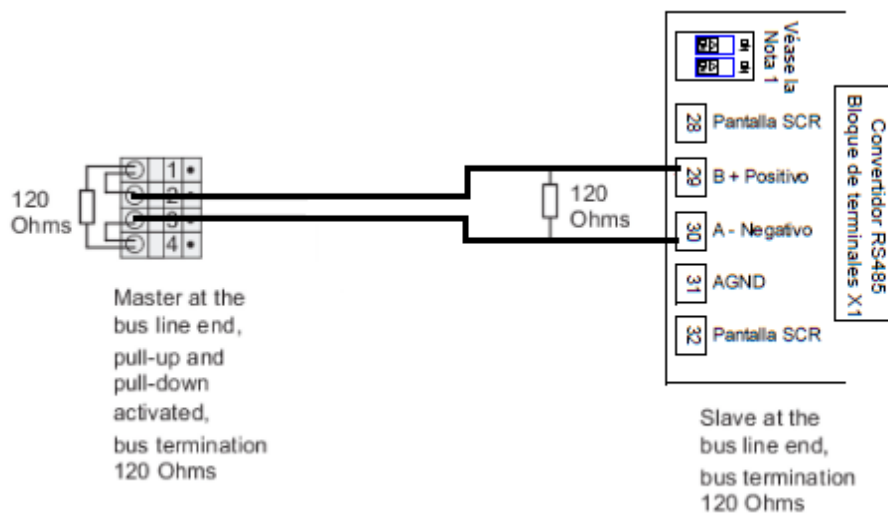


Figura 22 – Conexión RS485 PAC PM573-ETH-Convertidor ACS550 [28] [30]

En el convertidor se dispone de un “switch” para indicar al dispositivo que se va hacer uso del puerto de comunicación RS485. En la figura 23, se observa la posición para poder establecer la comunicación por dicho puerto.

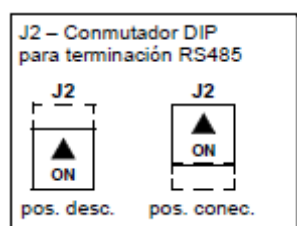


Figura 23 – Conmutador RS485 ACS550 [30]

6.4.3. Conexión Panel HMI-PAC

La conexión entre el panel HMI i el autómata programable se constituye a través del protocolo RS485 (Para más información sobre el protocolo véase apartado 6.4.3). En este caso, los dos dispositivos disponen de conector y no de bloque de terminales. Los conectores son DE-9 y se construye un cable con dos terminales de par trenzado y apantallado.

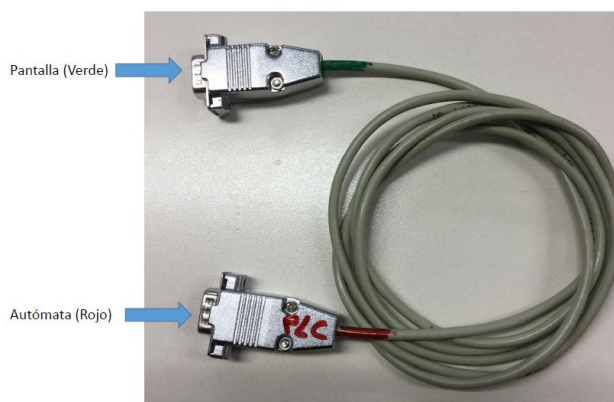


Figura 24 – Cable conxionado PM573-ETH COM2 – CP630

La conexión del autómata se realiza a través del COM2 y la correspondencia entre sus pins se observa en la figura 25:

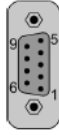
	1	FE	Functional earth	
	2	TxD	RS-232 Transmit Data	output
	3	RxD/TxD-P	RS-485 Receive/Transmit	positive
	4	RTS	RS-232 Request To Send	output
	5	SGND	Signal Ground	0 V supply out
	6	+5 V		5 V supply out
	7	RxD	RS-232 Receive Data	input
	8	RxD/TxD-N	RS-485 Receive/Transmit	negative
	9	CTS	RS-232 Clear to Send	input
	Shield	FE	Functional earth	

Figura 25 – Pins conxionado RS485 CP630 [28]

La topología final del esquema del cable se muestra en la figura 27. Donde podemos observar las resistencias empleadas para la conexión de cada dispositivo y la longitud máxima recomendada del cable.

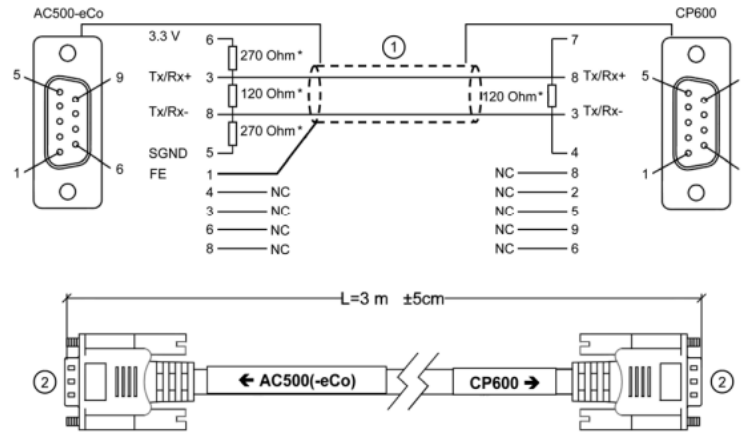


Figura 26 – Cable con terminales DE-9 conexión PM573-ETH – CP630 [12]

6.4.4. Cableado del sistema

La alimentación del sistema se realiza a través de una fuente de alimentación de 24V y 20W de Telemecanique, ahora propiedad de Schneider Electric. Esta fuente de alimentación suministrará la tensión de todo el sistema, alimentando el autómata, los módulos de expansión y el panel de control.

Los módulos de E/S del autómata tienen dos tipos de alimentación. Un bus dependiente de la base de conexiones que alimenta el funcionamiento del módulo en sí, y un sistema de alimentación independiente que suministra potencia a las salidas y entradas. Deben estar alimentadas las dos opciones, sino el autómata nos reportará un error, incluso si no se utilizan las E/S del módulo. (Véase apartado 7.2.9.2). A continuación, se presentan el esquema de conexionado para los módulos de expansión del sistema.

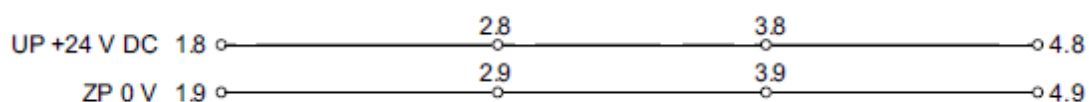


Figura 27 – Conexionado de alimentación módulos DA501 – CD522 [20] [25]

El esquema unifilar de conexionado del sistema completo es el siguiente:

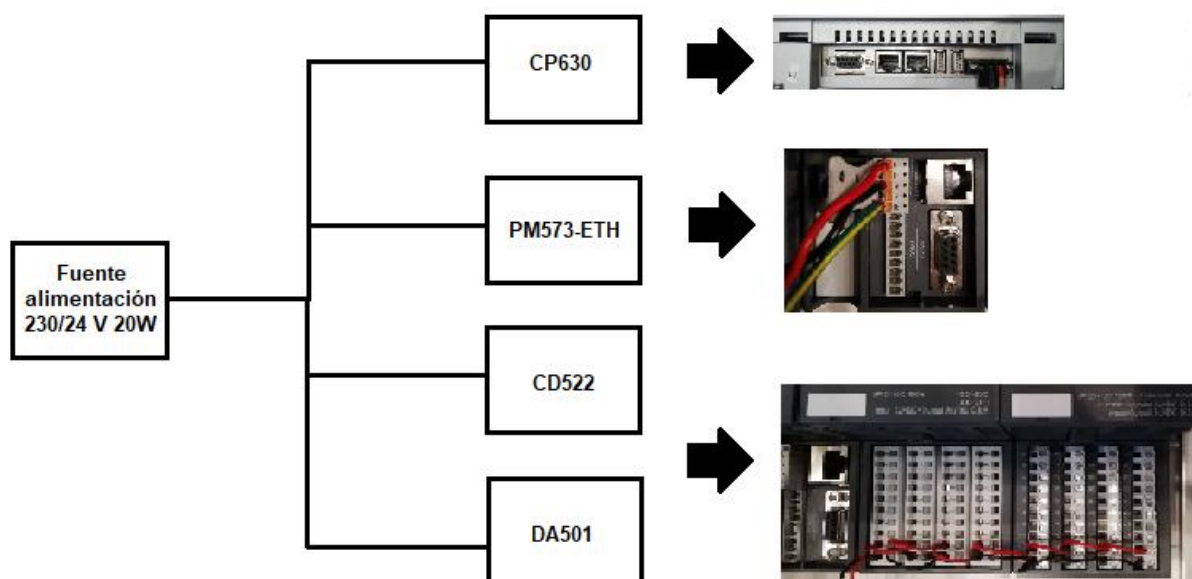


Figura 28 – Conexionado de alimentación del sistema

7. Software

En el presente proyecto se realiza el control de un convertidor a través de un autómata programable y de un panel de control HMI (véase el diagrama del sistema en el apartado 5). Para la realización y programación del control del sistema se van a emplear tres softwares: Automation Builder, CODESYS y Panel Builder 600.

Automation Builder nos permitirá la comunicación, gestión y configuración de los dispositivos del sistema. Este software está basado en el estándar IEC61131-3. Por este motivo se sustenta en el software de programación CODESYS. En este programa se realizará el control del variador a partir del lenguaje de programación de bloques funcionales. Es donde reside el código del sistema.

Finalmente, se realizará el diseño y la programación del panel de control a través del software Panel Builder 600. Desde donde se ejecutará la conexión y configuración del panel destinado al control HMI.

7.1. Normalización IEC61131-3

A día de hoy, la mayoría de fabricantes de sistemas de automatización, tienen sistemas de control específicos y propios, donde la programación está diseñada concretamente para su producto. Asimismo, existe una gran complejidad para la conexión entre distintos sistemas y fabricantes. Este hecho implica un coste elevado de los sistemas y una dependencia de la marca del fabricante. La normalización pretende flexibilizar este hecho.

El estándar IEC61131 es un conjunto de normas, aplicables a los autómatas programables y sus periféricos, así como herramientas de programación y depuración, diseñados para el control de procesos industriales y maquinaria. Estas normas han sido ratificadas de forma idéntica por UNE.

La norma no tiene el objetivo de regular el sistema automatizado, sino la seguridad y funcionalidad del autómata, el cual, es un componente básico en un sistema automatizado.

El estándar se divide en 5 partes, de las cuales, solo dos no tienen rango de norma en la actualidad, apartados 4 y 8.

- **Parte 1:** Visión general
- **Parte 2:** Hardware
- **Parte 3:** Lenguajes de programación
- **Parte 4:** Guías de usuario
- **Parte 5:** Comunicación
- **Parte 7:** Control “Fuzzy” o borroso
- **Parte 8:** Guías para el diseño, implementación y uso de lenguajes de programación.

El apartado IEC61131-3 de la norma tiene el objeto de regular la interface entre el usuario y el autómatas, es decir, el software y, más concretamente los lenguajes de programación. La norma define los lenguajes de programación más corrientes, las reglas sintácticas y semánticas, así como la ampliación y adaptación de los equipos.

Se puede diferenciar, claramente, entre dos apartados de la norma. En un primer lugar, los elementos comunes del código, y una parte posterior, donde se hace referencia específicamente a los lenguajes de programación. Los lenguajes de programación se tratan con más profundidad en el apartado CODESYS (Para más información véase el capítulo 7.3.2).

En los elementos comunes, el estándar define los tipos de datos y las variables, el modelo de software, la comunicación de datos, el modelo de programación y los elementos de configuración.

En los tipos de datos se estandarizan los de uso común como booleanos, enteros y reales, entre otros. En la parte dedicada a las variables define la asignación de estas como direcciones de hardware, memoria o datos. Estas variables pueden ser locales, si corresponden a un apartado concreto del programa, o globales, si se definen para ser utilizadas en la totalidad del programa.

La comunicación de variables define el flujo de datos dentro del programa, entendiendo también la relación entre variables, ya sean locales o globales y la comunicación entre bloques funcionales.

En el modelo de programación se hace referencia la estructura y procesado del programa. Se establecen las unidades de organización, llamadas POU's, donde se organiza el cuerpo del programa. Contienen la definición de variables y los recursos de programación,

tales como funciones, bloques o subprogramas. Todos estos elementos o conjuntos lógicos quedan recogidos, definidos y estructurados en este apartado de la norma.

En la última parte de la norma, se indican los elementos de configuración. Es en este apartado donde se recogen los cuatro componentes básicos: los recursos, las tareas, las variables globales y las vías de acceso.

7.2.Automation Builder: Suite integral para la automatización

Automation Builder es una “suite” de programación integrada que permite desarrollar proyectos para autómatas, paneles de control, convertidores, robots y PLC de seguridad de ABB desde un mismo entorno. Esto proporciona una configuración total del sistema y la portabilidad de todo el proyecto en un único archivo.

El software de ABB contiene todas las herramientas necesarias para la configuración, programación y depuración, a la vez que combina funciones de diagnóstico y simulación. El objetivo principal de desarrollo es el campo de la automatización en la industria.

Permite la posibilidad de programar en diversos lenguajes, incluso combinarlos dentro de una misma aplicación. Además de la utilización y configuración como servidor web.

Automation Builder se basa en el software de programación CODESYS, y por tanto, en los estándares de programación normalizada IEC-61131-3.

7.2.1. Instalación

En primer lugar, debemos descargar el software de la página web de ABB (www.abb.com/automationbuilder) y comprobar que nuestro hardware y sistema operativo cumple con los requisitos mínimos.

Los requisitos mínimos del sistema son los siguientes:

- Pentium PC, 1 GHz, 3 GB RAM
- Memoria de disco duro: 10 GB
- Tarjeta gráfica SVGA 256 colores, resolución 1024 x 768 Píxel

- Puerto USB 2.0
- Windows XP (Service Pack 3 mínimo), Windows 7 (32/64 bits) o superior

Una vez hemos descargado el software, iniciamos el archivo ejecutable y seleccionamos el idioma que utilizaremos tanto en la instalación, como posteriormente en el programa.

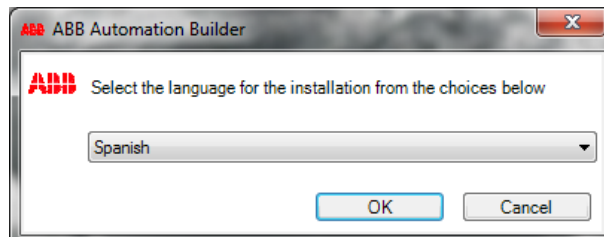


Figura 29 – Automation Builder. Proceso de Instalación 1

A continuación, seleccionamos iniciar la instalación y aceptamos el contrato de licencia.

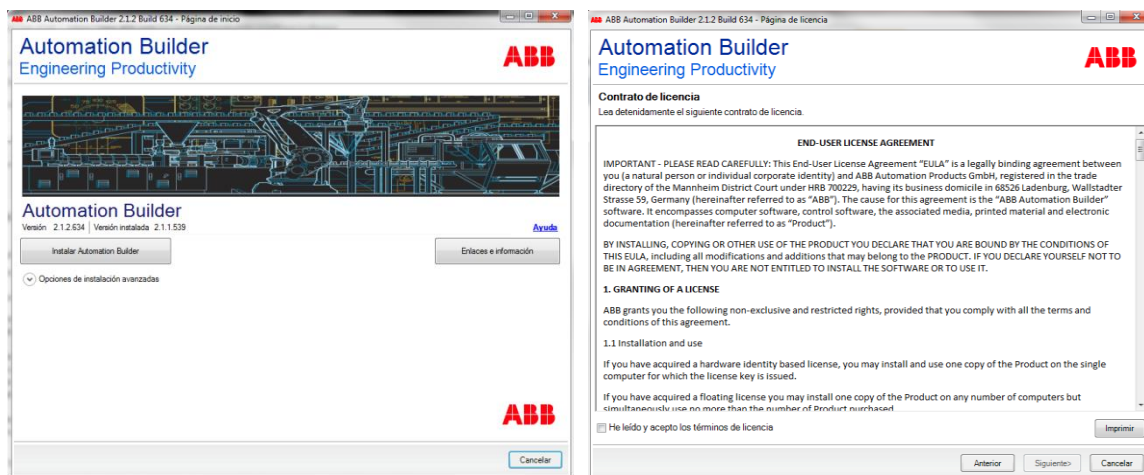


Figura 30 - Automation Builder. Proceso de Instalación 2

Posteriormente, seleccionamos el tipo de instalación, dependiendo de la clave de activación que tengamos. Es posible seleccionar entre Basic, Standard o Premium. Además, es posible seleccionar los paquetes de software individualmente, y realizar así una instalación personalizada según el uso y los objetivos de nuestro proyecto.. Finalmente, seleccionamos “Descargar e Instalar”.

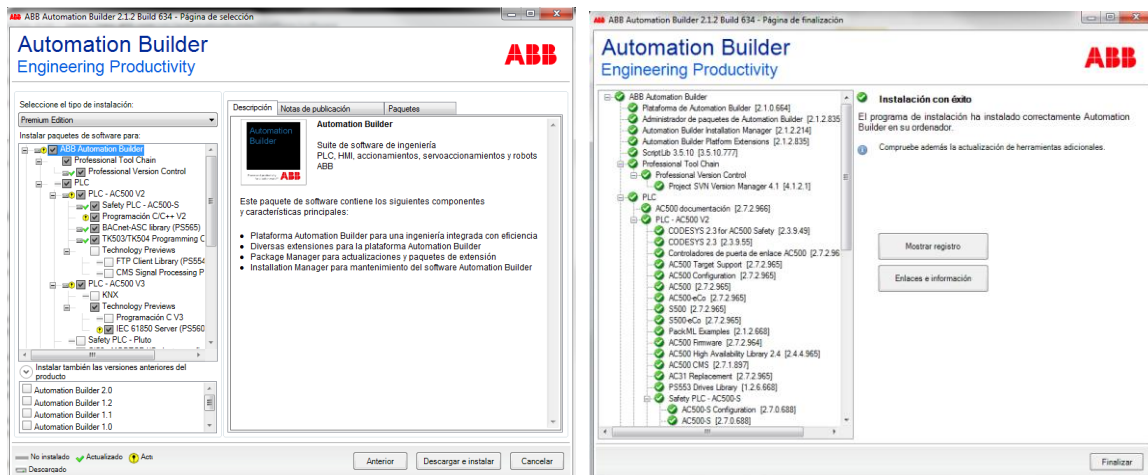


Figura 31 - Automation Builder. Proceso de Instalación 3

En último lugar, si la instalación se ha realizado correctamente se mostrarán todos los paquetes instalados con un “tic” de color verde y podremos clicar en “Finalizar”.

7.2.2. Explicación del entorno

En el presente apartado, se describe el entorno de Automation Builder. Se detallarán las funciones principales para poder desarrollar cualquier proyecto de automatización con entera autonomía.

El entorno del software se puede dividir en cinco partes diferenciadas. A continuación, se especificarán las funciones y opciones que ofrecen cada una de ellas. Algunas partes del menú desplegable, tienen botones que realizan la misma función en el menú de inicio rápido. Se pueden identificar con facilidad, ya que los iconos del menú desplegable y del inicio rápido son idénticos.

En el cuadro 1 de la figura 33, tenemos la barra de menú y los iconos de inicio rápido. Es la parte de gestión del programa, y desde ella, accederemos a las diversas funcionalidades del software. A continuación, se detallan algunas de funcionalidades más importantes de los diferentes menús.

- Archivo: es el menú de gestión de archivo del proyecto. Podremos crear un nuevo proyecto, guardarlo, o gestionar configuraciones de página e impresión.

- Edición: permite opciones de edición del documento, como deshacer, buscar o reemplazar.
- Ver: ofrece un acceso directo a distintas partes del software, como dispositivos, mensajes o volver a la página de inicio.
- Proyecto: permite la configuración del proyecto y realizar tareas relacionadas con el conjunto de los dispositivos. Estas tareas, también, se pueden realizar desde el árbol de dispositivos. Asimismo, encontraremos las opciones de Importar/Exportar, que nos permite generar archivos de distinto formato para otros softwares o dispositivos que no sean de ABB.
- En línea: es el menú de comunicación con el autómata. Des de él nos conectamos al autómata (iniciando sesión), guardamos el proyecto en el autómata o realizamos un Reset. Se dispone de botones homónimos en el menú de inicio rápido.
- Depuración: Permite el inicio del programa en el autómata. Antes de realizar este paso, es necesario estar conectado al autómata. También dispone de sus correspondientes botones en el inicio rápido.
- Herramientas: da acceso a las diferentes herramientas del software, entre ellas, podremos encontrar el "Installation Manager" para ampliar la instalación, la configuración de IP o el repositorio de dispositivos para actualizar los equipos disponibles.
- Ventana: se dispone de las opciones para operar con las distintas ventanas abiertas en la parte principal del programa. Básicamente, son opciones de visualización.
- Ayuda: en este apartado, tenemos el acceso a la ayuda de Automation Builder. Es un documento extenso y de gran utilidad. Dispone de contenido sobre el software Automation Builder y CODESYS, los distintos dispositivos de ABB, explicación de las diferentes herramientas de los dos software y ejemplos de

programación. En el menú también se dispone del enlace a varias páginas web de ABB, donde existen tutoriales y las últimas novedades del fabricante.

- IEC61850: corresponde con el estándar de automatización y control de subestaciones. Permite la conversión de código y la conexión con servidores.

En el cuadro 2 de la figura 33, tenemos el árbol de dispositivos. En él se disponen todos los dispositivos, comunicaciones y elementos del proyecto. Se puede acceder a las características de cada uno de ellos, o bien, añadir nuevos equipos.

En este árbol, también, tenemos incluido el programa del autómatas. Es el icono "Application", desde donde accederemos al programa CODESYS para iniciar al software de programación.

En la pestaña inferior del árbol de dispositivos, tenemos una segunda pestaña "POU". En ella podremos acceder a algunas configuraciones de programa que vamos a desarrollar. Estas características se configurarán automáticamente en CODESYS, si bien, también podemos configurarlas desde CODESYS.

En el cuadro 3 de la figura 33, tenemos la parte principal del programa. Está organizada en forma de pestañas, y aquí se nos abrirán las ventanas de los diferentes elementos del árbol de dispositivos. Es un software multitarea que permite tener abiertas distintas pestañas al mismo tiempo. Algunas pestañas en la parte izquierda disponen de un submenú, en el cual, se tienen acceso a algunas funciones adicionales.

En el cuadro 4 de la figura 33, se dispone de las comunicaciones del software. Aquí observaremos, los procesos de configuración, guardado o compilación. También, se nos comunicarán posibles errores, advertencias o mensajes. El cuadro está organizado en columnas y, por tanto, en las comunicaciones se nos indicará el objeto o la posición del error. Este sistema, permite la ubicación rápida de cualquier problema de compilado.

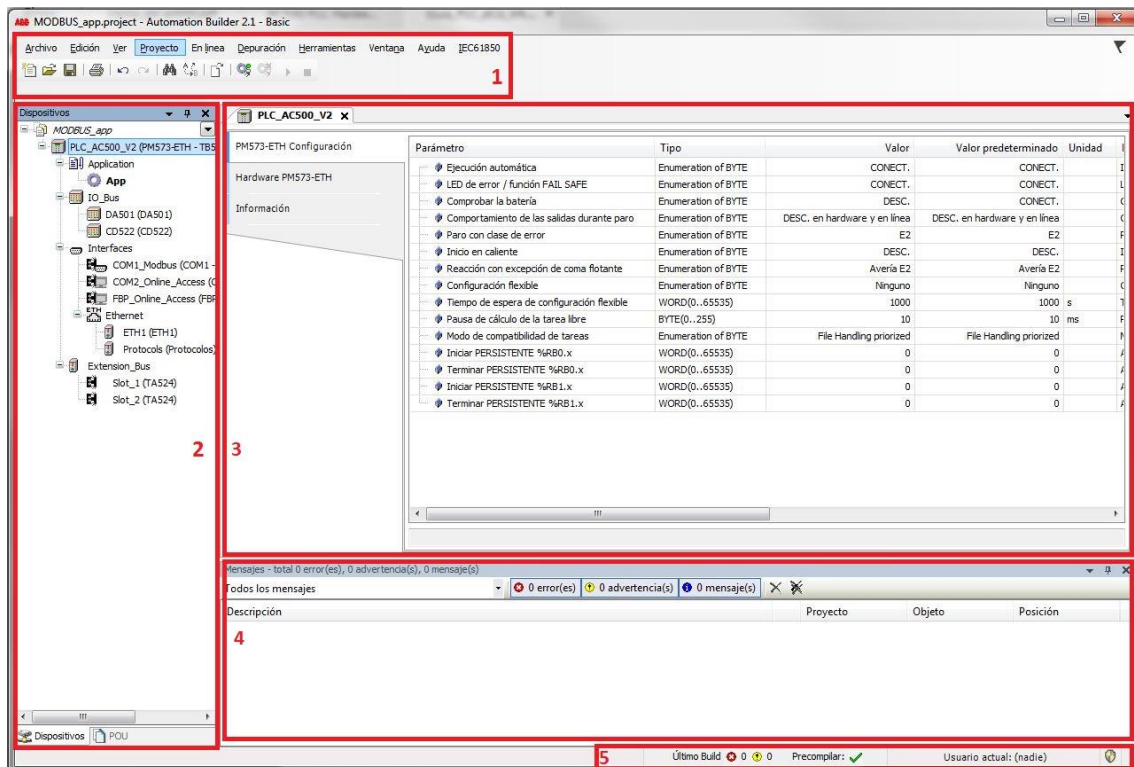


Figura 32 – Automation Builder. Entorno

En el cuadro 5 de la figura 33, tenemos un pequeño resumen, en forma de iconos, del estado del proyecto. Aquí podemos ver, el usuario logado, los errores y la fase que se encuentra el proyecto. En las fases veremos si el proyecto se encuentra compilado, si el autómatas está conectado, o bien, si el programa se ha iniciado y está en marcha.

7.2.3. Creación de un nuevo proyecto

Para la creación de un nuevo proyecto, iniciamos el software Automation Builder y en la pantalla principal nos dirigimos a la opción “Nuevo proyecto”. Si no nos encontramos en la pantalla principal, también nos podemos dirigir al menú “Archivo” y seleccionar “Nuevo proyecto”.

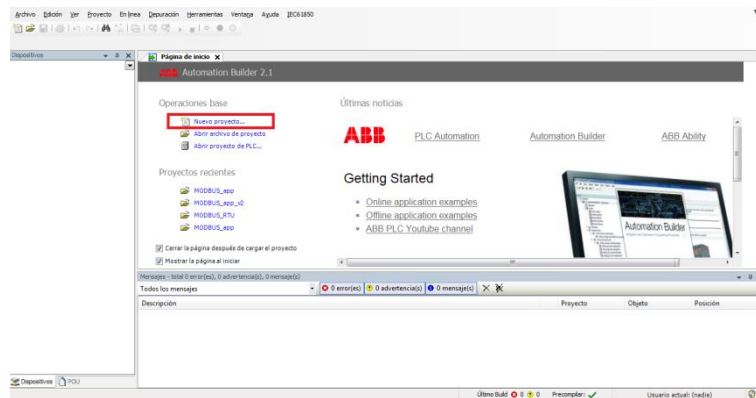


Figura 33 – Automation Builder. Creación nuevo proyecto 1

A continuación, se debe otorgar un nombre al proyecto y seleccionar la carpeta de destino, donde los archivos serán guardados. En esta ventana, podemos elegir entre un proyecto AC500 que da la opción a seleccionar el autómata, o bien, un proyecto vacío. En este último caso, los equipos del proyecto se seleccionan con posterioridad. En el presente proyecto se selecciona la opción “Proyecto AC500”.

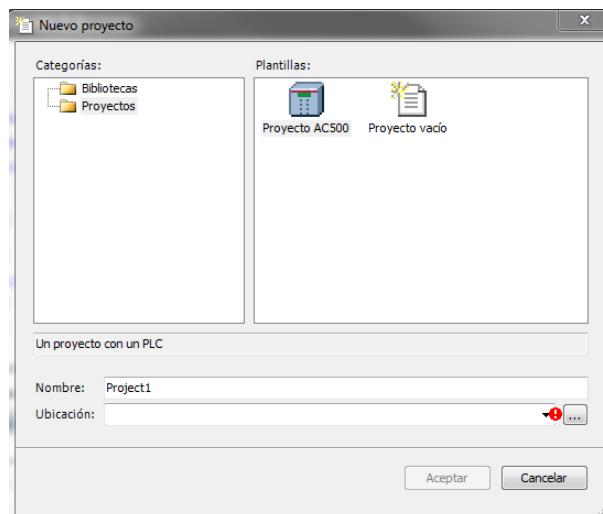


Figura 34 - Automation Builder. Creación nuevo proyecto 2

En la siguiente ventana, es preciso seleccionar el autómata del proyecto. Para ello nos movemos en el árbol de dispositivos en la parte izquierda de la ventana. Seleccionaremos la familia del autómata y en la parte derecha de la ventana se nos mostrarán los modelos disponibles. Se selecciona el modelo PM573-ETH y clicamos en “Agregar PLC”.

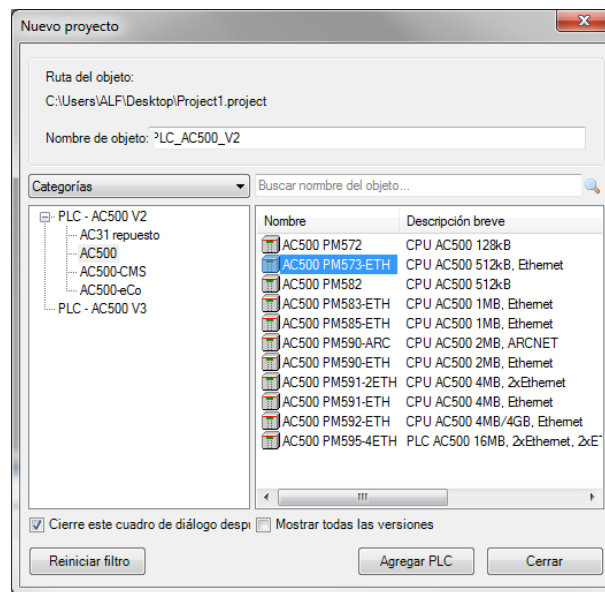


Figura 35 - Automation Builder. Creación nuevo proyecto 3

Con este proceso habremos finalizado la parte inicial de creación del nuevo proyecto y, por tanto, se nos abrirá la página principal del software. En la parte izquierda, se puede observar el árbol de dispositivos que hemos configurado.

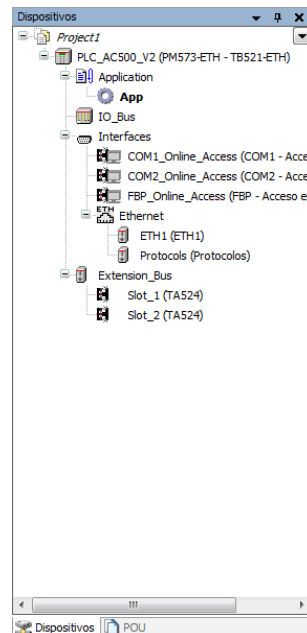


Figura 36 - Automation Builder. Creación nuevo proyecto 4

Una de las características de los PAC's de ABB es que son modulares. Es en este momento, donde se deben configurar los módulos de extensión para finalizar con la creación del nuevo proyecto.

Para configurar los módulos, nos dirigimos al árbol de dispositivos y con el botón izquierdo del ratón seleccionamos “IO_BUS”. En el desplegable, debemos dirigirnos a “Agregar objeto”.

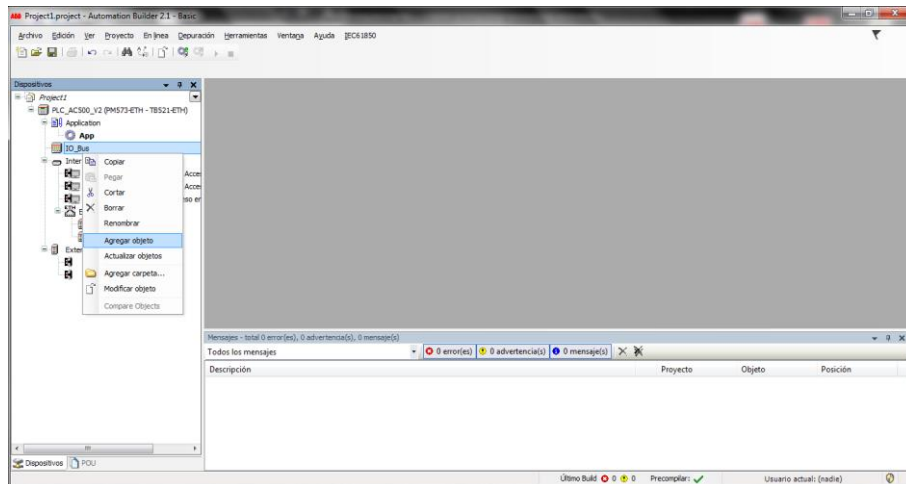


Figura 37 - Automation Builder. Creación nuevo proyecto 5

Se abrirá una ventana con el mismo formato anterior, es decir, cuando se ha seleccionado el autómata. La diferencia reside en el tipo de dispositivos que se encuentran en este cuadro de diálogo. Nos movemos por la familia de productos y seleccionamos el módulo que se desea añadir al proyecto. Se debe repetir el proceso por cada uno de los módulos que se deseen añadir.

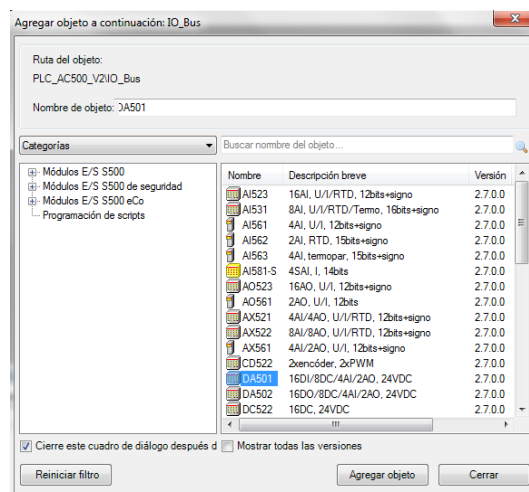


Figura 38 - Automation Builder. Creación nuevo proyecto 6

En el presente proyecto, se dispone de dos módulos de ampliación, el DA501 y el CD522. Una vez finalizado el proceso de añadir los dispositivos del sistema, el árbol de dispositivos queda de la siguiente forma:

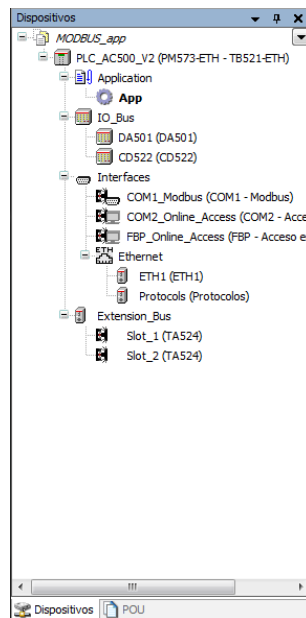


Figura 39 - Automation Builder. Creación nuevo proyecto 7

7.2.4. Configuración del autómata

Los autómatas programables de ABB disponen de parámetros configurables que permiten adaptar cada módulo al tipo de proyecto que se va a ejecutar. Todos los módulos tienen algún tipo de parametrización, la cual, se realiza del mismo modo. Algunos módulos tienen sus propias particularidades, como los módulos de E/S (Para ampliar la información de estos módulos véase apartado 7.2.6). En este apartado nos centraremos en la CPU, y veremos qué posibilidades nos ofrece Automation Builder.

Para acceder a la configuración de un módulo abriremos su pestaña. En este caso clicamos sobre “PLC_AC500_V2(PM573-ETH –TB521-ETH)” para acceder a la CPU. Observamos que en la pestaña se nos abren de tres opciones distintas en el submenú: configuración, hardware e información.

En el menú de configuración, tenemos una lista con todos los parámetros modificables del módulo en cuestión. Las columnas existentes nos permiten ver el tipo de dato y una descripción de cada parámetro.

Los parámetros se modifican en la columna de “Valor”, donde se nos abrirá un desplegable con las opciones disponibles, o bien, podremos introducir el valor numérico requerido. En la columna siguiente, tendremos la posibilidad de observar el valor por defecto. Esto nos permite conocer si algún parámetro ha sido modificado.

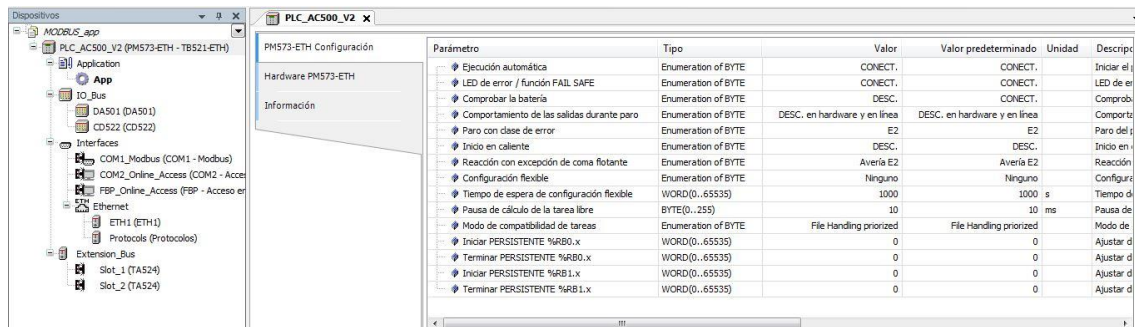


Figura 40 – Automation Builder. Configuración automática 1

La siguiente opción es la de Hardware. En este apartado podremos modificar el modelo de dispositivo. Esto puede resultar interesante si tenemos un proyecto que queremos ejecutar en dos modelos de autómatas diferentes o hemos actualizado la instalación. Por tanto, nos permite modificar el dispositivo sin tener que volver a realizar todo el proyecto desde el inicio.

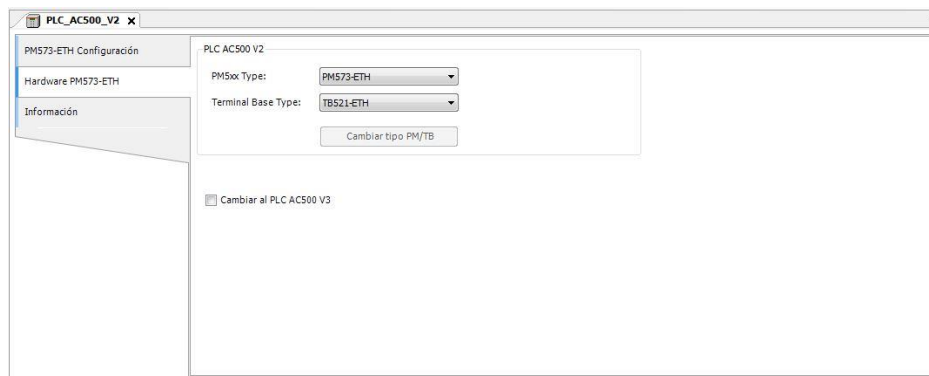


Figura 41 - Automation Builder. Configuración automática 2

Finalmente, en la opción de información es posible consultar las características del módulo. También tendremos acceso a la guía de instalación. Asimismo, tenemos disponible un link de ayuda con un extenso temario sobre el dispositivo y sus posibilidades de programación.

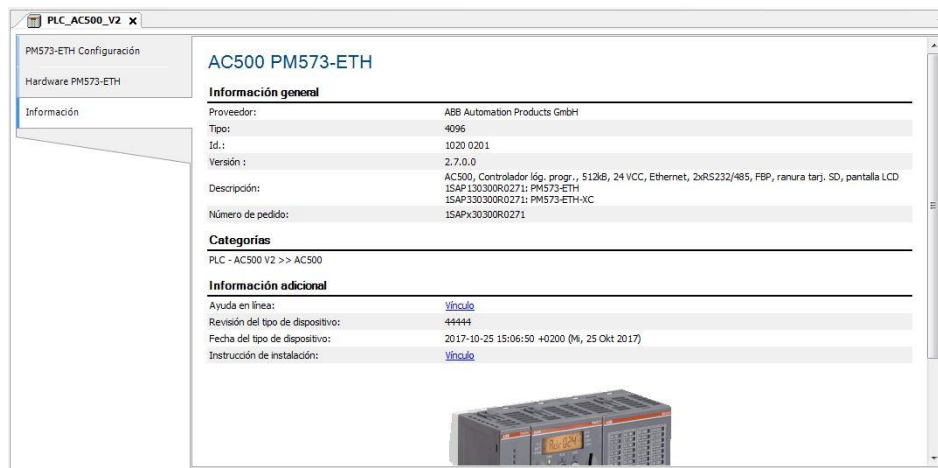


Figura 42 - Automation Builder. Configuración automática 3

7.2.5. Configuración IP

La conexión entre el automático y el PC se realiza a través de Ethernet. Por tanto, se deben configurar los parámetros de comunicación. En este sentido, deberemos especificar las características de nuestra red local y en el software Automation Builder, las propias del automático. En primer lugar, se configurarán las correspondientes IP's, para posteriormente comprobar la comunicación del software de programación.

La dirección IP del PC debe estar en el mismo rango que la IP de la CPU del automático. Esto quiere decir que los tres primeros parámetros de la dirección deben coincidir, del modo 192.168.3.X. Asignando después el valor X con la dirección individual de cada dispositivo. La máscara de subred debe permanecer en 255.255.255.0. Se asignarán las siguientes IP's a cada dispositivo:

Equipo	IP	Mascara de subred
PC	192.168.3.249	255.255.255.0
PAC	192.168.3.10	255.255.255.0

Tabla 8 – Configuración IP's comunicación PC /PAC

Para modifica la IP del PC nos dirigimos a Windows, y en el panel de control, seleccionamos “Conexiones de red y recursos compartidos” que se encuentra en el menú de

“Redes e Internet”. En el icono de “Conexión de área local” clicamos con el botón izquierdo y seleccionamos “Propiedades”. En la ventana que se nos abre buscamos el elemento “Protocolo de internet versión 4 (TCP/IPv4)” y seleccionamos propiedades. Elegimos la opción de “Usar la siguiente dirección IP” e ingresamos el valor anteriormente indicado. Para finalizar, aceptamos el proceso.

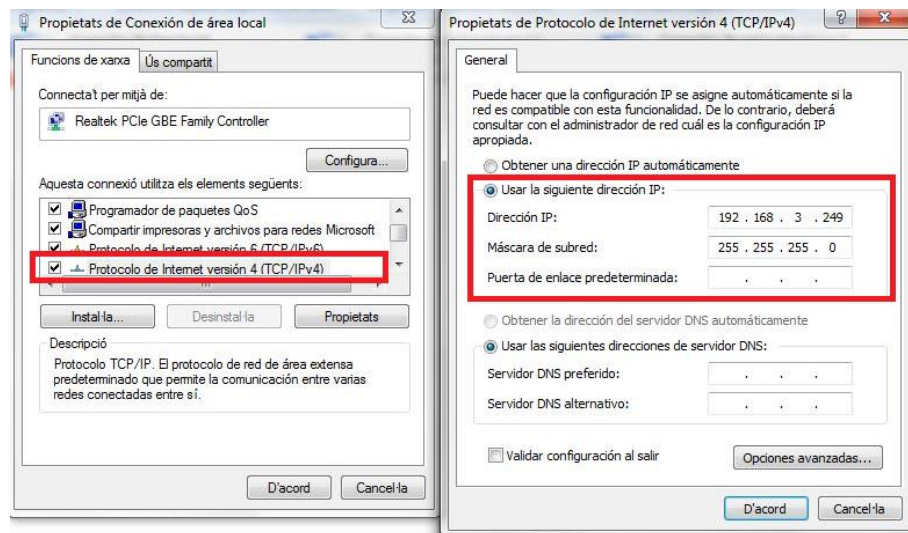


Figura 43 – Automation Builder. Configuración IP 1

A continuación, se debe asignar la IP al autómatas. En el árbol de dispositivos, nos dirigimos al elemento “Ethernet” y seleccionamos “ETH1(ETH1)” para abrir la pestaña. En la nueva pestaña seleccionamos “Configurar IP”.

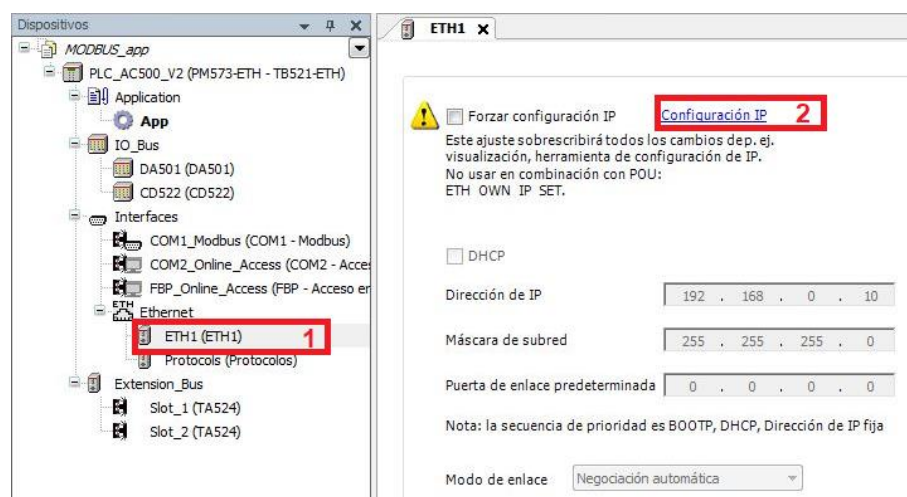


Figura 44 - Automation Builder. Configuración IP 2

En la nueva pestaña, clicamos en el botón “Scan”. En este proceso, el software realiza una búsqueda de los dispositivos conectados y nos muestra su dirección de IP. Cabe recordar que, para realizar este proceso el autómata debe estar encendido y conectado al PC mediante el correspondiente cable Ethernet.

Una vez se ha detectado el autómata, lo seleccionamos y en la parte inferior de la pestaña introducimos los parámetros de conexión que tenemos en la tabla 8. Finalizamos el proceso clicando en “Send Configuration”.

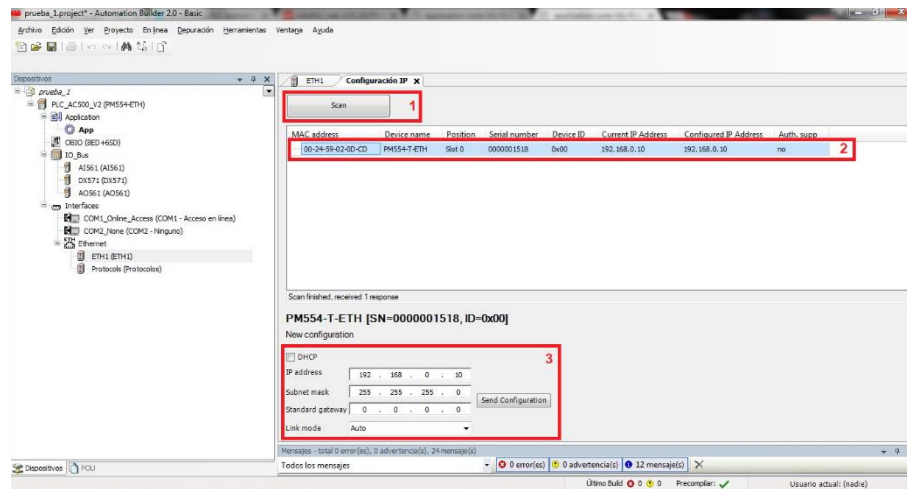



Figura 45 - Automation Builder. Configuración IP 3

Una vez se ha finalizado todo el proceso, podremos conectar el software con el autómata mediante el botón  de la barra de tareas. Si el proceso se ha realizado correctamente el software pasará a modo “online”.

7.2.6. Asignación de variables

El software Automation Builder permite la asignación de variables a los módulos configurados en el proyecto. Estas variables serán declaradas como globales en el software de programación CODESYS y, por tanto, podrán ser utilizadas en cualquiera de las POU's existentes en el proyecto. Incluido también otros sistemas dentro del proyecto como programaciones HMI o robots. La asignación de variables puede ser a partir de variables de nueva creación, o bien, variables ya existentes, el software las detectará automáticamente.

Para la creación de variables deberemos abrir la pestaña del módulo donde queremos asignar las variables. En la ventana del dispositivo, encontraremos tres opciones de menú: Configuración, Asignación E/S y Lista de mapeado E/S.

En el menú “Configuración” podemos modificar algunas características del módulo. Si nos centramos en las variables, en este menú podremos configurar las características de cada canal. Por ejemplo, podremos asignar un canal como entrada o salida, prestación importante de los autómatas de ABB. Asimismo, se deberá asignar el rango de tensión o corriente en el que queremos trabajar (-10...+10 V, 12 bits + signo, 0...10 V, 0...20mA, 4...20mA). Para configurar el valor del canal nos dirigimos al menú configuración, y en la E/S que se quiere utilizar se abre el desplegable determinando el valor deseado.

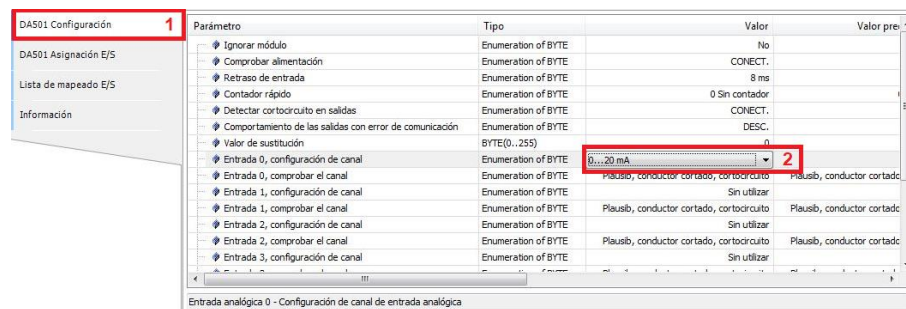


Figura 46 - Automation Builder. Asignación de variables 1

A continuación, seleccionaremos la opción del menú Asignación E/S. En este apartado asignaremos las variables. Debemos abrir el desplegable de la dirección donde queremos asignar la variable y en la columna de “Asignación” introducir el nombre de dicha variable. Como se ha comentado con anterioridad, en el caso de ser una variable ya creada, en otro módulo por ejemplo, el software lo detectará automáticamente.

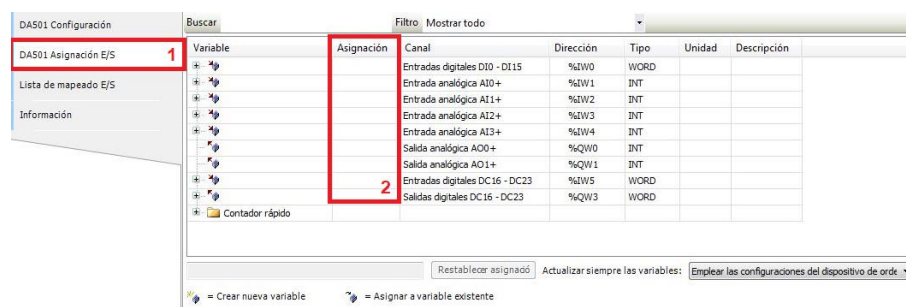
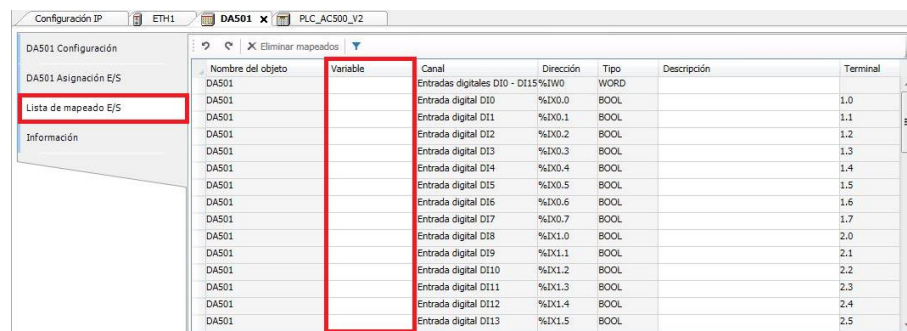


Figura 47 - Automation Builder. Asignación de variables 2

Finalmente, en la opción “Lista de mapeado E/S” tenemos un listado único de todos los canales del módulo, sin el filtrado anterior por dirección. En este apartado, también podremos asignar variables en la columna de “Variable”, además de añadir una descripción. En este apartado es, interesante la columna “Terminal”. En ella se nos indica, la dirección física del canal en el módulo.



Nombre del objeto	Variable	Canal	Dirección	Tipo	Descripción	Terminal
DA501		Entradas digitales D10 - D15 %IW0		WORD		
DA501		Entrada digital D10	%IX0.0	BOOL		1.0
DA501		Entrada digital D11	%IX0.1	BOOL		1.1
DA501		Entrada digital D12	%IX0.2	BOOL		1.2
DA501		Entrada digital D13	%IX0.3	BOOL		1.3
DA501		Entrada digital D14	%IX0.4	BOOL		1.4
DA501		Entrada digital D15	%IX0.5	BOOL		1.5
DA501		Entrada digital D16	%IX0.6	BOOL		1.6
DA501		Entrada digital D17	%IX0.7	BOOL		1.7
DA501		Entrada digital D18	%IX1.0	BOOL		2.0
DA501		Entrada digital D19	%IX1.1	BOOL		2.1
DA501		Entrada digital D110	%IX1.2	BOOL		2.2
DA501		Entrada digital D111	%IX1.3	BOOL		2.3
DA501		Entrada digital D112	%IX1.4	BOOL		2.4
DA501		Entrada digital D113	%IX1.5	BOOL		2.5

Figura 48 - Automation Builder. Asignación de variables 3

7.2.7. Bibliotecas

El software Automation Builder tiene el objetivo de integrar todos los dispositivos de un proyecto de automatización en un mismo software. Para ello, dispone de bibliotecas adicionales. Estas bibliotecas generan nuevos recursos de programación para dispositivos como convertidores, robots o sistemas de control predefinidos. Sin embargo, algunas de estas bibliotecas están destinadas únicamente para productos ABB. Normalmente, tienen el fin de facilitar la comunicación entre dispositivos, a la vez que facilitan la programación de los mismos.

Las bibliotecas se pueden descargar desde el “Installation Manager” en el menú “Herramientas”. Este instalador de paquetes permite ampliar la instalación inicial del software. No obstante, algunas librerías específicas de programación no se encuentran en este instalador y deben descargarse directamente de la página oficial de ABB.

En el presente proyecto, se realiza el control de un convertidor mediante Modbus RTU. Para el control de convertidores de ABB tenemos disponible la librería “PS553-DRIVES”. Esta librería contiene bloques funcionales que permiten la comunicación directa mediante diferentes protocolos, el control básico y el acceso a los parámetros del variador.

El proceso para la utilización de las librerías es semejante para todas. En primer lugar, descargamos la biblioteca “PS553-DRIVES” de la página de ABB. El archivo es un “.zip” que debemos instalar iniciando el ejecutable “setup.exe”. En este proceso es importante, seleccionar la carpeta de destino de la nueva librería, ya que el programa que realizaremos buscará esa ubicación. Si modificamos el destino de la carpeta, el programa nos indicará un error y no será posible abrir CODESYS para la tarea de programación.

Una vez hemos instalado la biblioteca debemos cargarla en el software CODESYS. Para abrir CODESYS hacemos doble-clic en “Application”.

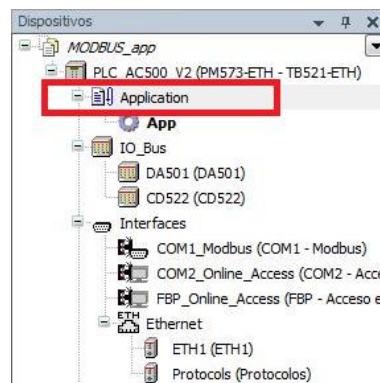


Figura 49 - Automation Builder. Instalación de bibliotecas paso 1

Cuando se haya iniciado CODESYS, nos dirigimos a la pestaña “Resources”. En esta pestaña se puede configurar y organizar el proyecto. Seleccionamos la opción “Library Manager” y en la nueva ventana aparecerán las bibliotecas cargadas. En esta ventana, clicamos con el botón izquierdo y seleccionamos “Additional library”. En el explorador de archivos, buscamos la carpeta “PS553-DRIVES” y seleccionamos las tres librerías que nos aparecen. Cada una de ellas tiene bloques distintos, desde el control hasta la comunicación Modbus RTU o Modbus TCP (esta última no será utilizada en el presente proyecto).

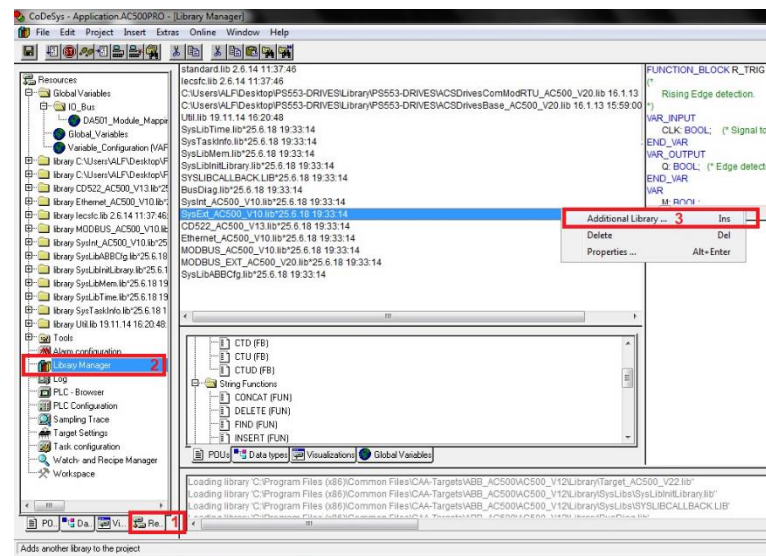


Figura 50 - Automation Builder. Instalación de bibliotecas paso 2


Finalmente, observaremos que los archivos han sido incluidos en el árbol de bibliotecas de nuestro proyecto. A partir de este momento, las nuevas funcionalidades podrán ser utilizadas en la programación del editor (Para ver las nuevas funcionalidades véase el apartado 7.3.4).

7.2.8. Actualización firmware autómeta

La compañía ABB continúa evolucionando y mejorando sus productos, incluso una vez estos ya se encuentran en el mercado. Esto permite tener un autómeta seguro, más eficiente y con nuevas aplicaciones.

Esta evolución de los productos se muestra periódicamente, en una actualización de firmware. Si no se realizan estas actualizaciones, el autómeta puede no conectarse correctamente con el software y existir incompatibilidades que no permitan la descarga del programa.

Existen dos maneras de actualizar el firmware del autómeta. La primera es instalando el propio software desde una tarjeta SD. Este caso, está destinado para infraestructuras que carecen de acceso a internet. La segunda manera es a través del software Automation Builder y la correspondiente conexión a internet.

Para actualizar el firmware desde el software, se debe conectar el autómata al software Automation Builder. Para ello seleccionaremos el botón “Iniciar” en la barra de menús, el icono indicado es . El software realiza, automáticamente, una comprobación del firmware del dispositivo. En caso de encontrar una versión más actualizada, nos mostrará el mensaje que se observa en la figura 51. Seleccionaremos “Mostrar la información de la versión” y clicamos en continuar.

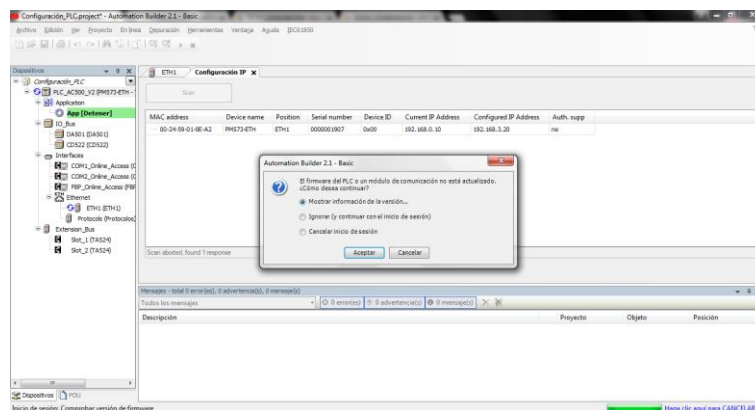


Figura 51 - Automation Builder. Actualización firmware paso 1

En la nueva pestaña, podremos observar que firmware del dispositivo se encuentra desactualizado. Se mostrará en un icono rojo con una cruz en la barra de estado. A continuación, para iniciar la actualización se debe seleccionar el botón “Actualización Firmware”.

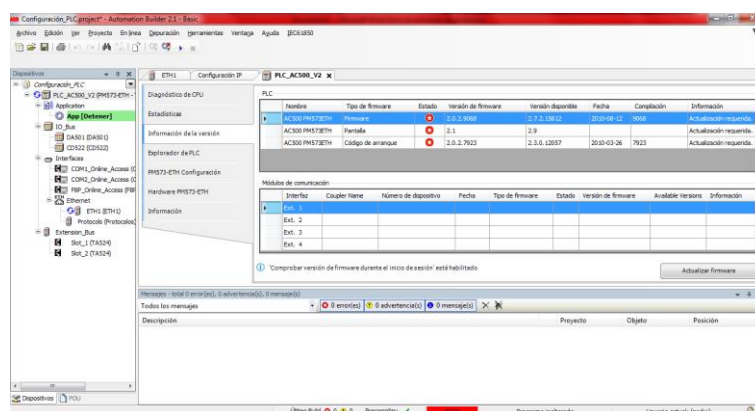


Figura 52 - Automation Builder. Actualización firmware paso 2

Se nos indicara el estado del proceso en una nueva ventana, desde donde se puede seguir la evolución del proceso. Para asegurarnos que la conexión es correcta y la actualización se está desarrollando de manera satisfactoria, debemos comprobar que los indicadores del autómatas “RUN” y “ERROR” parpadean de manera constante durante este proceso.

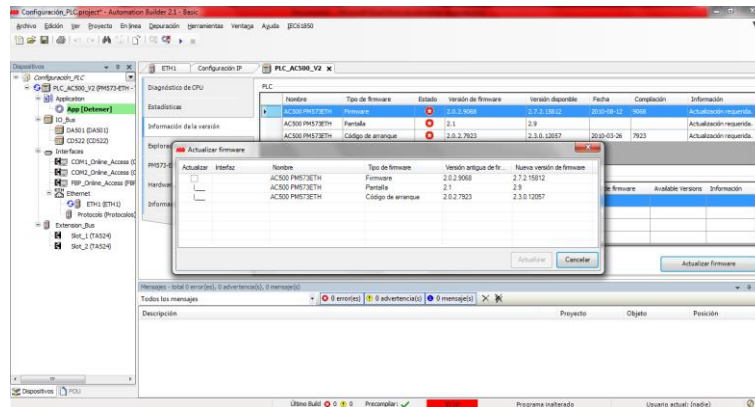


Figura 53 - Automation Builder. Actualización firmware paso 3

Finalmente, el software vuelve a la pestaña inicial. Si el icono de estado es verde, el proceso se habrá completado correctamente. En otras ocasiones, desde esta misma pestaña podremos consultar la versión de nuestro firmware y la fecha de actualización. Para acceder a ella, debemos seleccionar el autómatas en el árbol de dispositivos y dirigirnos a la opción de firmware, siempre con la sesión iniciada.

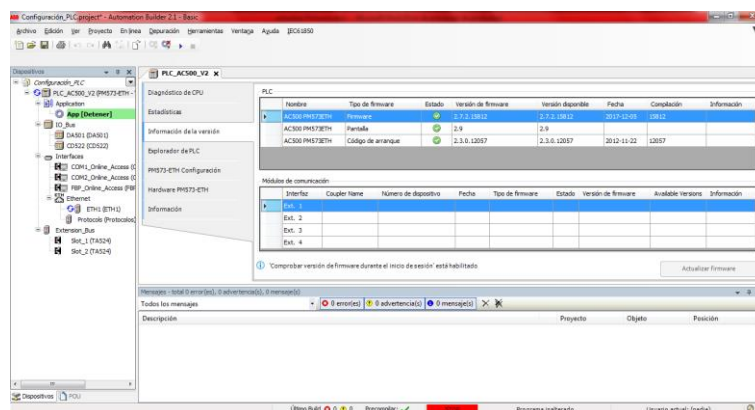


Figura 54 - Automation Builder. Actualización firmware paso 4

7.2.9. Errores software

La primera vez que conectamos un autómata al software Automation Builder, nos pueden aparecer algunos errores recurrentes. Estos errores pueden deberse a prestaciones del propio autómata, a las características del mismo o incluso a un mal dimensionamiento de la instalación.

Durante el proceso de creación de este proyecto, se han encontrado algunos errores que se han ido solventando a partir del avance en el conocimiento de la tecnología de ABB. En los capítulos posteriores se detalla la solución a dos errores comunes en la conexión del autómata. Estos errores pueden impedir la compilación del programa o la descarga en el autómata y, por tanto, es de vital importancia solucionarlos para poder continuar con el desarrollo de nuestro sistema.

7.2.9.1. Borrar errores del autómata

Los errores del autómata pueden resetearse de dos maneras, desde el teclado de control del propio autómata, o bien, desde el software Automation Builder.

Para resetear el autómata desde el propio dispositivo, en primer lugar, debemos visualizar los distintos errores con las teclas de posición. Los errores solo se pueden resetear de uno en uno. Es un sistema de seguridad para no comprometer la instalación al obviar un error grave.

Una vez tenemos el error en la pantalla del *"display"*, se debe teclear *"VAL"* y posteriormente *"OK"* para confirmar la operación. Este proceso deberá repetirse por cada error del autómata.

Para resetear el autómata desde el software Automation Builder deberemos estar conectados al autómata, por tanto, con la sesión iniciada. A continuación, abrimos la pestaña de la CPU haciendo doble clic en *"PLC_AC500_V2(PM573-ETH –TB521-ETH)"* y nos dirigimos a *"Diagnósticos de CPU"*. En la pantalla se mostrarán los distintos errores del autómata. Para actualizar la lista de errores se debe presionar *"Leer errores"*. Finalmente, para resetear el autómata deberemos clicar en el botón *"Borrar todos los errores"*. Si los errores no persisten, el autómata volverá al estado de *"stand-by"*.

En este caso si existe la posibilidad de eliminar todos los errores de una vez, ya que la visualización en forma de lista permite ver todos los errores al mismo tiempo.

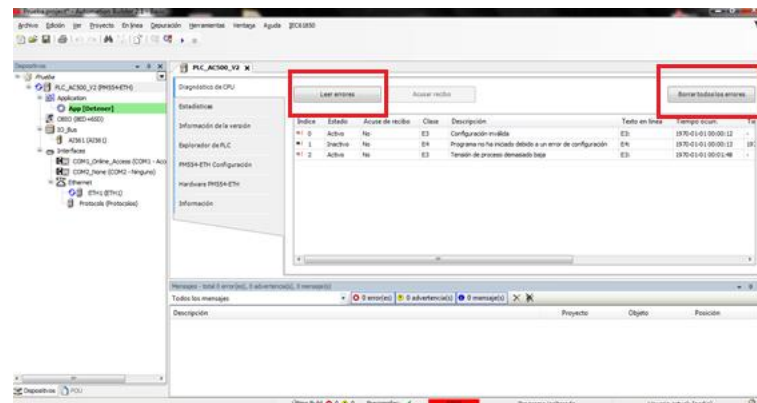


Figura 55 - Automation Builder. Borrar errores del autómata

7.2.9.2. Error batería auxiliar

Algunos modelos de autómatas ABB tienen dentro de sus prestaciones el reloj en tiempo real (RTC). Es el caso del modelo PM573-ETH del presente proyecto. El reloj en tiempo real permite programar aplicaciones a una hora o día concreto. Así como desarrollar procesos que no se sustenten sobre contadores internos.

El reloj en tiempo real necesita una fuente de energía incluso cuando el autómata se encuentra apagado. Por este motivo, los autómatas que tienen RTC disponen de una ranura para baterías o pilas, normalmente, de litio. Si no se dispone de esta pila, o se encuentra agotada, el software no permitirá descargar el programa en el autómata. Esto es por motivos de seguridad, ya que pueden haberse programado procesos que se inicien en momentos incorrectos, o procesos esenciales que no se lleven a cabo.

Si el programa del autómata no hace uso del RTC y no se dispone de la batería, es posible desactivar esta opción. Para realizar esta acción, debemos abrir la pestaña del autómata en el árbol de dispositivos. En la venta del autómata seleccionamos la opción configuración.

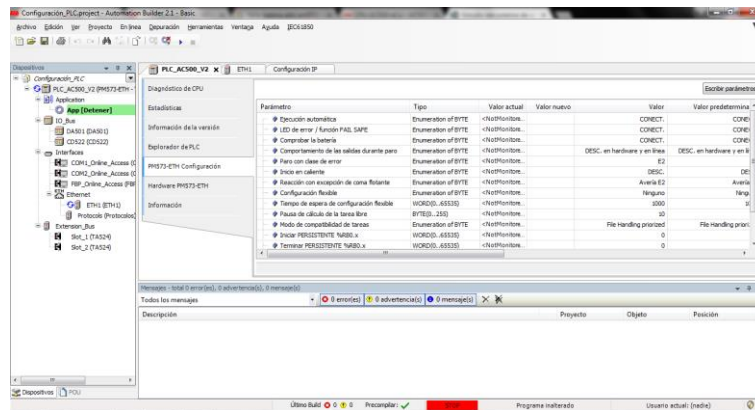


Figura 56 - Automation Builder. Error batería auxiliar 1

En las opciones del menú de configuración, es preciso dirigirse a la opción “Comprobar batería”. En la columna de valor seleccionamos “Desactivar”. Una vez realizado este proceso, el software nos permitirá descargar el programa en el autómatas sin disponer de la batería para el reloj en tiempo real.

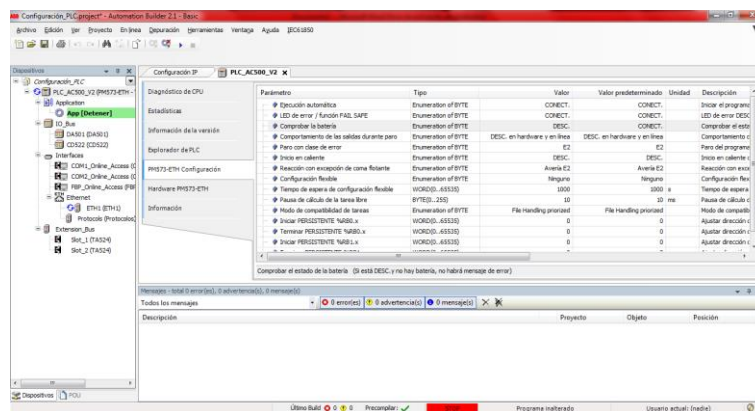


Figura 57 - Automation Builder. Error batería auxiliar 2

7.2.9.3. Tiempo de espera actualizar datos

El error actual nos permite compilar, a diferencia del anterior. Sin embargo, no nos permite iniciar la aplicación. En el instante de iniciar, nos indica un error que puede aparecer en dos versiones distintas. Puede mostrar el mensaje “Tiempo de espera actualizar datos”, o bien, como se puede observar en la figura 59, “configuración invalida” y “Tensión de proceso demasiado baja”.

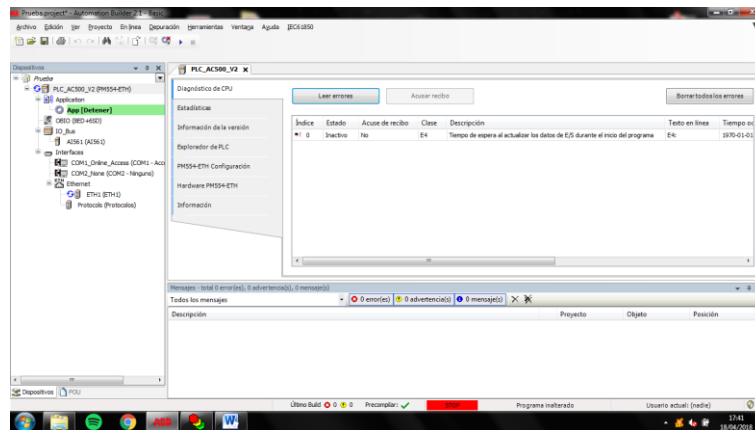


Figura 58 - Automation Builder. Error tiempo de espera actualizar datos 1

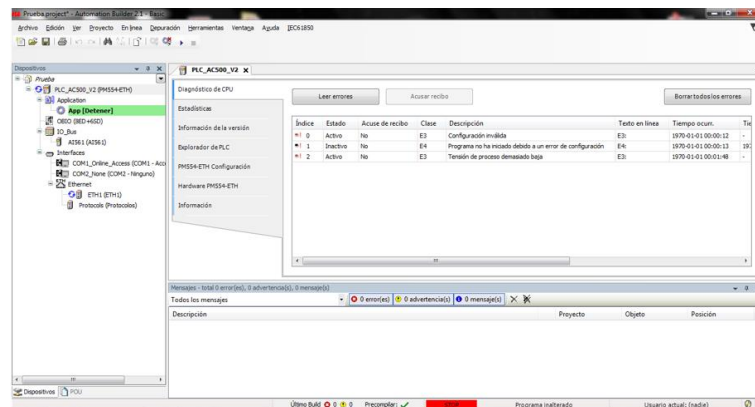


Figura 59 - Automation Builder. Error tiempo de espera actualizar datos 2

Estos errores indican una falta de potencia en la instalación del autómata. Puede deberse a distintos factores. En primer lugar, se debe comprobar el dimensionado de la fuente de alimentación. La fuente debe de ser capaz de alimentar no solo al autómata, sino a los módulos de ampliación, así como las conexiones de E/S de estos módulos si no disponen de una alimentación independiente.

Si se ha comprobado que la fuente de alimentación es correcta, se deberá revisar el cableado de la instalación. Un falso contacto en la alimentación de alguno de los módulos, puede generar este error, incluso cuando el led de “RUN” está encendido. Esto se debe a que la alimentación principal del módulo, se produce a través del bus que se encuentra en la base de terminales. Sin embargo, el sistema de E/S puede estar sin tensión.

7.3.CODESYS: plataforma de programación

CODESYS es un acrónimo de *Controller Development System* (sistema de desarrollo del controlador). Es un software de programación diseñado para cubrir todas las necesidades de los proyectos de automatización actuales. Es uno de los sistemas de programación estándar IEC61131-3 (Véase capítulo 7.1.) más utilizados. Está destinado principalmente a la programación de autómatas y reguladores industriales, no obstante, también permite la creación de código para microprocesadores.



Figura 60 – Icono CODESYS
CODESYS oficial web page. Recuperado de: <https://www.codesys.com/>

El software CODESYS permite la programación en los cinco lenguajes estándar. Además, dispone de un sistema de visualizaciones semejante a un SCADA interno para el control del programa desarrollado. Así como, un sistema de simulación offline para el desarrollo y detección de errores.

CODESYS es un software de código abierto y se puede descargar gratuitamente desde la página del fabricante. Es un programa versátil que da la posibilidad de trabajar con dispositivos de distintos fabricantes y combinarlos entre ellos.

Es una plataforma modular que permite la integración de bibliotecas o ampliaciones de software según la funcionalidad de cada proyecto. Así como, la adaptación a sistemas ya existentes o estándares como OPC y protocolos de comunicación diversos. En las últimas versiones se añaden características para la integración y programación de objetos IoT.

7.3.1. Explicación del entorno

En el presente apartado, se describe el entorno de CODESYS. Se detallarán las funciones principales para poder desarrollar la programación del autómatas en este entorno.

El entorno del software se puede dividir en cuatro partes diferenciadas. Seguidamente, se especificarán las funciones y opciones que ofrecen cada una de los apartados. Algunas partes del menú desplegable, tienen botones que realizan la misma función el menú de inicio rápido. Se pueden identificar con facilidad, ya que los iconos del menú desplegable y del inicio rápido son idénticos.

En el cuadro 1 de la figura 62, tenemos la barra de menú y los iconos de inicio rápido. Es la parte de gestión del programa, y desde ella, accederemos a las diversas funcionalidades del software. A continuación, se detallan algunas de funcionalidades más importantes de los diferentes menús.

- File: dispone de la gestión del archivo. Es un menú capado, ya que muchas de las funcionalidades se disponen desde Automation Builder.
- Edit: proporciona herramientas de edición, como deshacer, copiar o la navegación por los errores del programa.
- Project: permite gestionar el proyecto de programación. Aquí se realizan la compilación o la parametrización de las opciones del proyecto. También, dispone de sistemas de exportación/importación del proyecto, incluso para sistemas no reglados por el estándar IEC61131-3.
- Insert: dispone de los distintos elementos, mayoritariamente gráficos, para insertar en el programa. Estas acciones se pueden realizar, del mismo modo, desde el menú de inicio rápido.
- Extras: dispone de algunas opciones no vinculantes del programa. Mayoritariamente, para la organización y gestión del usuario.

- Online: proporciona las opciones de conexión con el autómata. Desde este menú se inicia o descarga el programa, además de acceder al modo simulación.
- Windows: opciones de navegabilidad del programa. Así como, disposición de ventanas o herramientas de visualización.
- Help: acceso al documento de ayuda. Contiene ejemplos de programación o links de información y proyectos de libre acceso.

En el cuadro 2 de la figura 62, tenemos un menú con cuatro pestañas. En la pestaña “POUs”, disponemos de los distintos programas del proyecto. En la pestaña “Data Types”, podremos crear tipos de variables no estándar o comunes. En la pestaña visualizaciones, disponemos de los diferentes programas visuales relacionados con las POU del proyecto. Finalmente, en la pestaña de “Resources” disponemos de las opciones del autómata y del proyecto para realizar las distintas configuraciones. La mayoría de estas opciones están capadas o se realizan desde Automation Builder, que las configura automáticamente.

En el cuadro 3 de la figura 62, tenemos el editor de programación. En la parte superior, se encuentra el espacio destinado a la declaración de variables. En la parte inferior, el documento de programación.

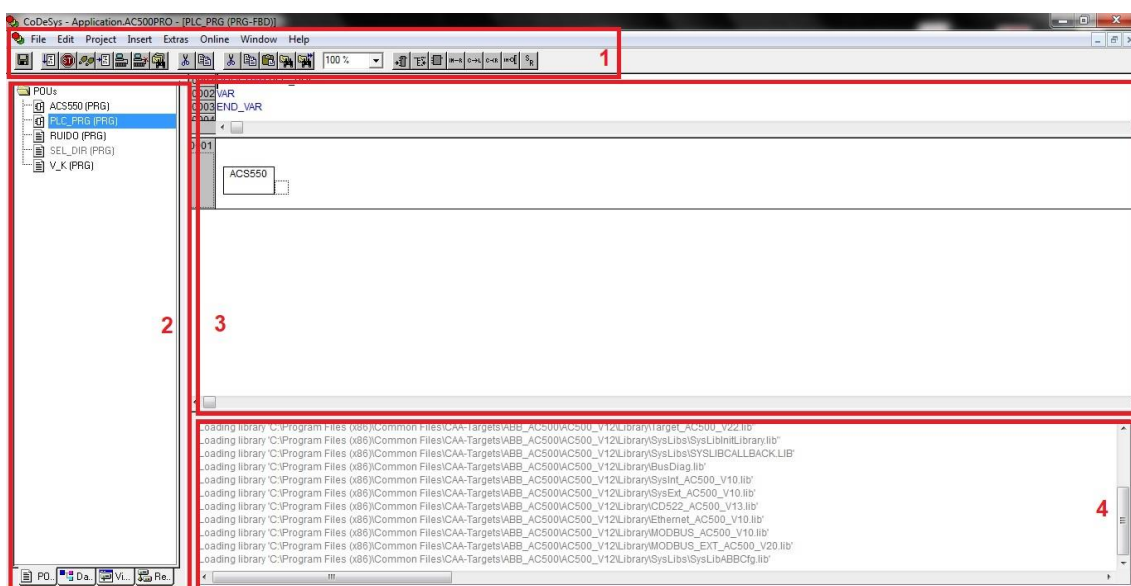


Figura 61 – CODESYS. Entorno

En el cuadro 4 de la figura 62, aparecen las comunicaciones del software. Aquí se observa el proceso de compilado, las acciones realizadas o los errores de programación.

7.3.2. Lenguajes de programación

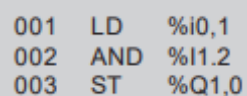
El software CODESYS permite la programación en seis lenguajes diferentes, de los cuales, cinco son los normalizados por el estándar IEC61131-3, norma que define la sintaxis y la semántica de los lenguajes (Para más información véase apartado 7.1). Cada POU (*Program Organization Unit*) del programa se relaciona con las demás, pero es independiente, y por tanto, pueden ser empleados distintos lenguajes en un mismo programa.

Los lenguajes de programación pueden ser clasificados entre gráficos o literales, según el estándar IEC61131-3. CODESYS admite la programación con los siguientes:

Lenguajes literales: formados por letras, números y símbolos.

- Lista de instrucciones (IL)

El lenguaje IL es una serie de expresiones lógicas y booleanas escritas secuencialmente. Tiene una estructura en forma de lista que consta de diversas instrucciones, donde cada fila es una instrucción independiente. La estructura de cada línea debe seguir la misma, y consta de: una etiqueta o número de instrucción, un operador y un operando.



```
001 LD %I0.1
002 AND %I1.2
003 ST %Q1.0
```

Figura 62 - CODESYS. Lenguajes de programación estándar (IL) [10]

- Texto estructurado (ST)

Es un lenguaje de nivel alto, basado en PASCAL y C. Es una construcción sintáctica que al ser evaluada aporta un valor. Está compuesta de operadores y operandos. Es un lenguaje estructurado en bloques y utilizado para el desarrollo de expresiones complejas. Permite la ejecución condicional, en bucles o funciones. En CODESYS es posible realizar un bloque con lenguaje ST,

para ser guardado y llamado posteriormente en un POU programada con un lenguaje distinto.

```

IF Dato = "EOF" THEN
  FOR indice:=1 TO 128 DO
    x:= Read_dato(Canal[indice]);
    IF x>2500 THEN Alarma:=TRUE;
  END_IF;
END_FOR;
END_IF

```

Figura 63 - CODESYS. Lenguajes de programación estándar (ST) [10]

Lenguajes gráficos: formados por figuras simbólicas.

- Diagrama de bloques funcionales (FBD)

Es un lenguaje simbólico representado por estructuras en forma de bloques. Estos bloques representan un operando lógico o una función interna compleja. Las variables de entrada se combinan dando paso a las de salida. Las entradas y salidas se conectan a los bloques mediante enlaces.

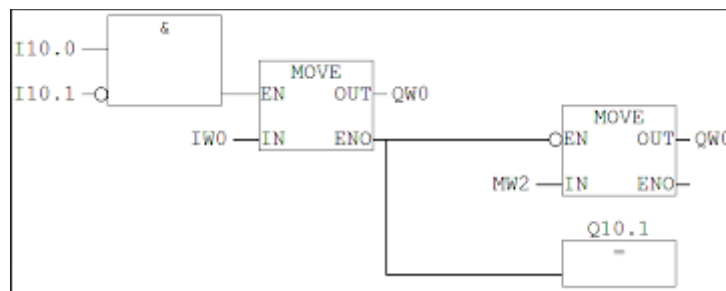


Figura 64 - CODESYS. Lenguajes de programación estándar (FBD) [10]

- Diagrama de contactos (LD)

Es un lenguaje basado en la lógica de contactos. Compuesto por barras de alimentación donde se representan las variables lógicas con el uso de bobinas y relés. Los contactos son normalmente abiertos o cerrados. Es posible la utilización de temporizadores, contadores e incluso la combinación con elementos complejos de bloques funcionales.

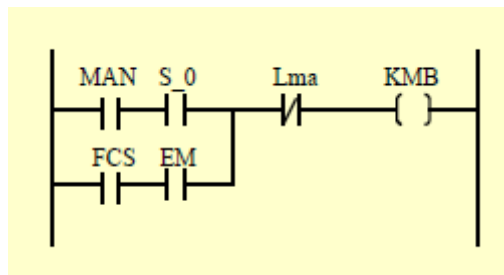


Figura 65 - CODESYS. Lenguajes de programación estándar (LD) [10]

- Bloques de función secuencial (SFC)

Lenguaje basado en Grafcet. Estructurado en las distintas acciones que se deben realizar a través del tiempo, que se asocian a diferentes pasos o transiciones. Permite una visión completa del programa, al tiempo que descompone cada acción o proceso en pequeñas partes. Al igual que el lenguaje de bloques funcionales cada componente puede ser programado en los distintos lenguajes, incluido el mismo SFC.

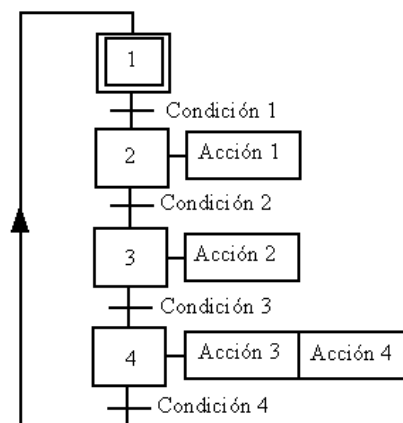


Figura 66 - CODESYS. Lenguajes de programación estándar (SFC) [10]

- Continuous Function Chart (CFC) – No incluido en el estándar IEC61131-3

Es una variación del lenguaje de bloques funcionales (FBD). Los bloques son predefinidos para la realización de aplicaciones secuenciales y repetitivas. Destinado a procesos sencillos y personal no especializado en lenguajes de programación. Este lenguaje no es compatible con los demás, y por tanto, no es posible una traducción automática.

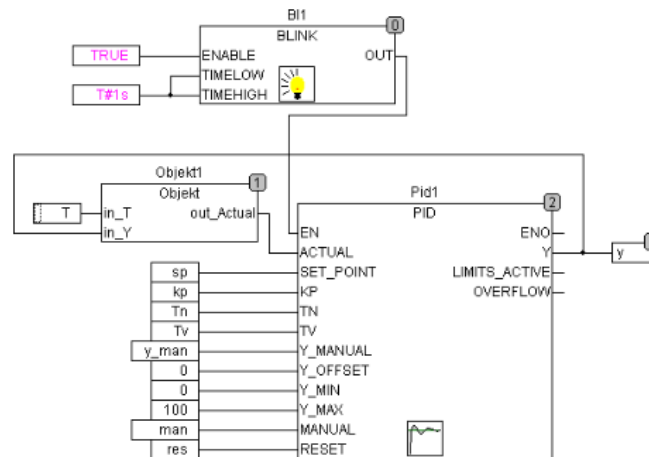


Figura 67 - CODESYS. Lenguajes de programación estándar (CFC) [10]

7.3.3. Definición de variables

La definición de variables en CODESYS puede ser global o local, tal y como se indica en el estándar IEC61131-3. Las variables locales se definen en cada POU y solo pueden emplearse en esa POU donde han sido definidas. En cambio, las variables globales se definen para todo el programa y pueden ser utilizadas en todas las POU's.

Cuando definimos una variable para una E/S en Automation Builder, estas se definen automáticamente en CODESYS como variables globales, y por tanto, podrán ser utilizadas en cualquier POU.

Si un proyecto necesita exportar las variables para ser introducidas en otro software de programación, es necesario declarar las variables como globales. En caso contrario, no aparecerán en el archivo de exportación y, en consecuencia, no podrán ser utilizadas por otros dispositivos. Este hecho se debe, principalmente, al uso restringido para cada POU que no permitiría la correcta comunicación de las variables destinadas, únicamente, a memorias internas en la POU de origen.

La definición de variables se realiza sintácticamente del mismo modo, tanto si son globales como locales. Se deberá introducir el nombre de la variable, seguido de dos puntos, y a continuación, el tipo de variable. Finalizando la sentencia con punto y coma. En caso de querer otorgar un valor a dicha variable, antes del punto y coma escribiremos dos puntos e igual seguido del valor. El tipo de variable siempre se debe escribir en mayúsculas.

Observaremos que el tipo de variable ha sido introducido correctamente porque el texto cambiará a color azul.

- **Ejemplo** → MARCHA:BOOL:=0;
 (Declaración de variable identificada como MARCHA de tipo booleano con valor inicial 0)

Las variables locales se introducen en el POU correspondiente. Cada POU tiene dos partes diferenciadas. En la parte inferior tenemos el cuerpo del programa, donde se desarrolla el código del programa y, la parte superior, donde se introducen las variables. Las variables deben declararse con la sintáctica anteriormente descrita.

```

0001 PROGRAM V_K
0002 VAR_INPUT
0003     EN:BOOL;
0004     EN2: BOOL;
0005     INPUT:INT;
0006 END_VAR
0007
0001 VEL:=INPUT;
0002 IF EN=TRUE THEN
0003     OUTPUT:=5000;
0004 ELSIF EN2=TRUE THEN
0005     OUTPUT:=15000;
0006 ELSIF EN=FALSE AND EN2=FALSE THEN
0007     OUTPUT:=VEL;
0008 END_IF
0009

```

Figura 68 - CODESYS. Declaración de variables 1

Para la declaración de variables locales, CODESYS permite otro método más dinámico en los lenguajes de programación gráficos. Cuando introducimos un elemento gráfico, las variables de entrada o salida necesarias aparecerán con tres símbolos interrogativos de color rojo. Si no posicionamos sobre ellos y apretamos "INTRO", nos aparece un cuadro de dialogo. En este cuadro de dialogo podremos declarar la variable. Nos permitirá escoger el tipo de variable, el nombre, el valor, la dirección asociada a un módulo o incluso introducir un comentario.

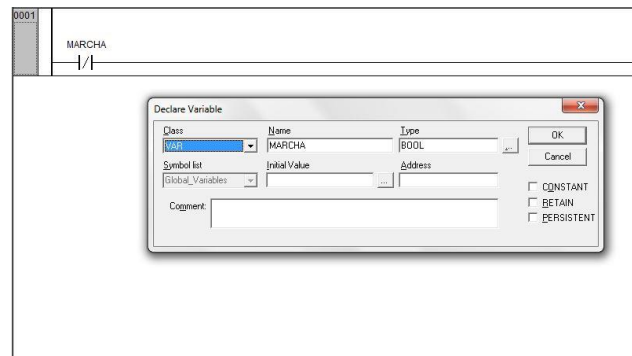


Figura 69 - CODESYS. Declaración de variables 2

Para la declaración de variables globales debemos dirigirnos a la pestaña de “Resources” y en ella seleccionar la opción “Global_Variables”. Se abrirá una nueva pantalla, con un formato semejante al apartado de declaración de variables de cada POU. En esta pantalla se declararán las variables que serán utilizadas en todas las partes del programa.

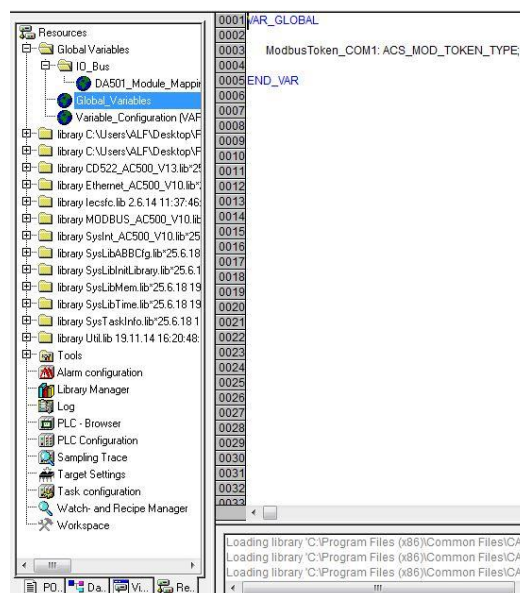


Figura 70 - CODESYS. Declaración de variables 3

Finalmente, una vez tenemos las variables declaradas podemos asignarlas de forma fácil y cómoda sin necesidad de recordarlas. Para ello, nos posicionamos en el lugar donde debe introducirse la variable. Como se ha comentado anteriormente, estará indicado con tres símbolos de interrogación rojos. Una vez estamos posicionados, apretamos “F2”. Se nos abrirá un cuadro de dialogo, en el cual, se encuentran todas las variables del proyecto y podremos

seleccionarla de manera fácil. Este sistema resulta interesante, ya que nos puede evitar errores de programación al introducir datos con errores gramaticales

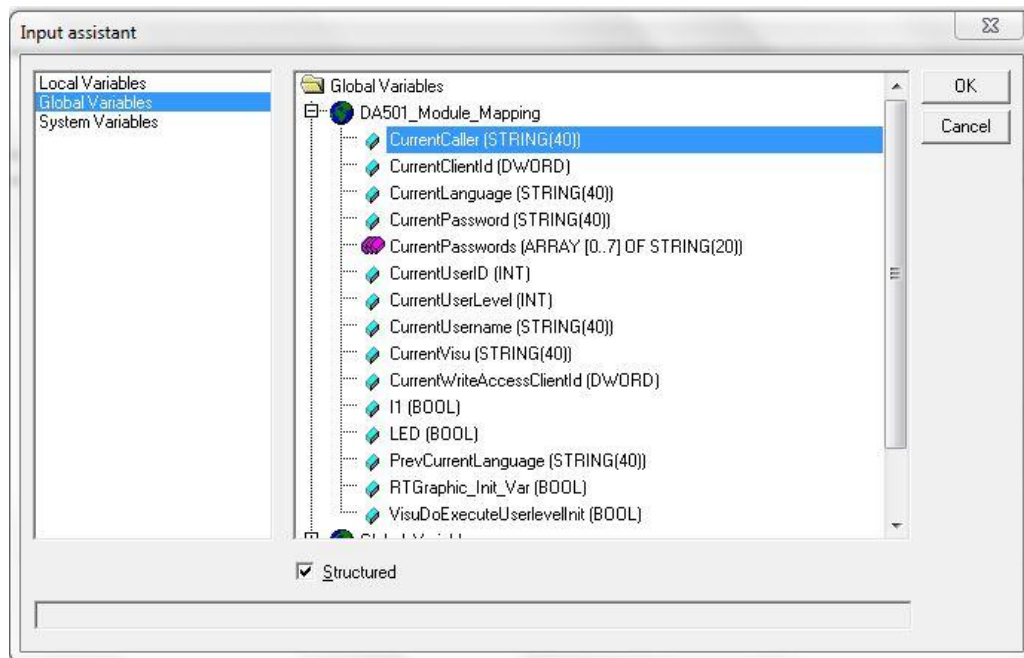


Figura 71 - CODESYS. Declaración de variables 4

CODESYS permite crear archivos de variables globales, lo cual, facilita la exportación de las variables necesarias en otros dispositivos. En el presente proyecto, se ha creado un archivo de variable globales para su posterior traslado al software Panel Builder.

Para la creación de este archivo, nos dirigimos a la pestaña de “Resources” y en la opción “Global Variables” clicamos con el botón izquierdo del ratón para seleccionar “Add Object”. En la venta de dialogo asignamos el nombre de dicho archivo.

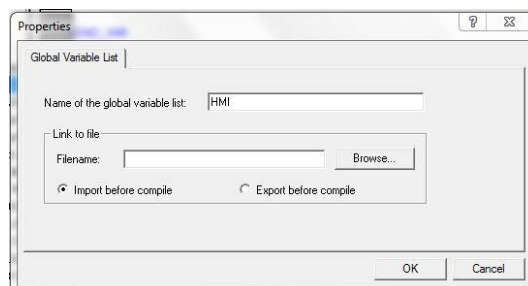


Figura 72 - CODESYS. Declaración de variables 5

El nuevo archivo es un editor de variables semejante al de variables globales. Por tanto, podrán ser utilizadas en todo el programa y nos permite ordenar las variables de la manera que requiera las necesidades de nuestro proyecto.

Si las variables se van a exportar para ser utilizadas por un bus de campo, estas deben asignarse a una memoria interna. Para ello, después del nombre de la variable introduciremos la palabra “AT” seguido del tipo de memoria y dirección.

Existen diferentes memorias para cada tipo de variable, y dependerá básicamente de la información a almacenar, por lo tanto, tendremos memorias de un bit o de ocho bytes entre otros. La declaración de una variable con la dirección de memoria incluida se realiza de la siguiente forma:

- **Ejemplo** → `MARCHA AT%MX0.0.1:BOOL:=0;`
(Declaración de variable identificada como MARCHA de tipo booleano con valor inicial 0 y dirección de memoria de un bit en la posición 0.1)

7.3.4. Programación en FBD

En el presente proyecto se han combinado, básicamente, dos tipos de lenguajes: bloques funcionales y lenguaje estructurado. La mayor parte del código del proyecto corresponde al lenguaje de bloques. El lenguaje estructurado se ha empleado, únicamente, para la creación de bloques personalizados para la realización de alguna tarea concreta. En este apartado, se introducirán las claves básicas para la programación en CODESYS, así como la parametrización y utilización del lenguaje de bloques.

Antes de iniciar el software CODESYS, deberemos guardar las modificaciones o parametrizaciones realizadas en Automation Builder. Para ello, nos dirigimos en el árbol de dispositivos al icono de “Application”, y hacemos clic con el botón izquierdo. En las opciones que nos aparecen seleccionamos “Crear datos de configuración”. Este proceso genera en CODESYS toda la configuración del proyecto, desde el tipo de autómatas hasta las variables declaradas en los módulos de E/S.

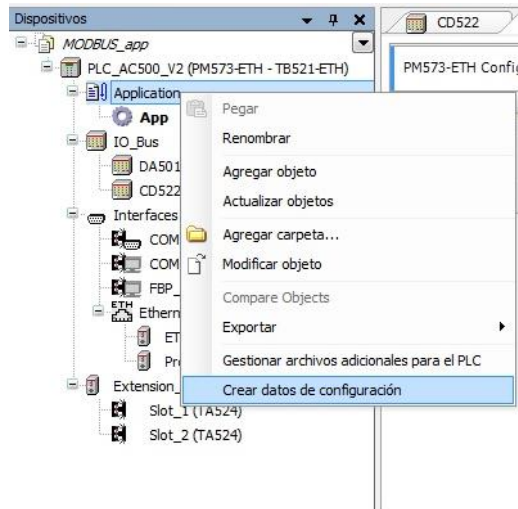


Figura 73 - CODESYS. Programación en FBD 1

Una vez realizado el proceso anterior, para entrar en CODESYS debemos hacer doble clic en el mismo icono de “Application”. Una vez cargado los datos necesarios, el software se iniciará.

CODESYS crea de manera predeterminada una POU principal llamada “PLC_PRG(PRG)”. Esta es la POU principal del programa y la que se iniciará en primer lugar cuando el autómata se encuentre compilando. No es conveniente realizar el código de nuestro programa en la POU principal, ya que si el autómata detecta un error invalidará la totalidad del programa. En cambio, si fraccionamos nuestro programa en diversas POU’s, una afectación en una de ellas no comprometerá la fiabilidad de las otras.

Creación de una nueva POU

Para la creación de una nueva POU nos dirigimos al árbol de POU’s situado en la parte izquierda del programa. Hacemos un clic con el botón izquierdo del ratón y seleccionamos “Add Object”. En el cuadro de dialogo que se nos abre deberemos seleccionar el tipo de POU, en este caso, seleccionamos “Program”. A continuación, se debe dar un nombre al nuevo POU. Finalmente, se seleccionará el lenguaje de programación que desea emplearse.

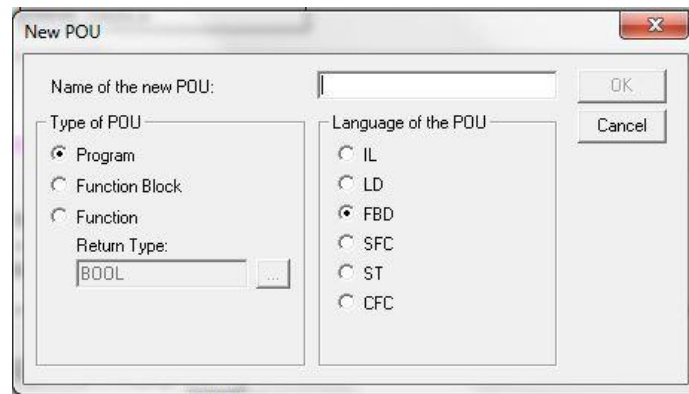


Figura 74 - CODESYS. Programación en FBD 2

Si se ha realizado correctamente el proceso, la nueva POU aparecerá en el árbol de POU's y podremos acceder a ella haciendo un doble clic.

CODESYS nos permite modificar el lenguaje de una POU ya creada. Esto creará una traducción automática del lenguaje, con el cual, ha sido programada. Solo es posible realizar este proceso con lenguajes compatibles. Los lenguajes que permiten esta conversión son LADDER, FBD y IL. Para realizar este cambio se debe hacer clic con el botón izquierdo del ratón en la POU que se desea modificar. Seguidamente, seleccionar "Convert Object". Finalmente, en la nueva ventana de dialogo abierta seleccionar el lenguaje de conversión.

Introducción de un bloque funcional

Para la introducción de un bloque funcional debemos entrar en la POU configurada con este lenguaje. Es posible que algunas POU's configuradas con otros lenguajes permitan la inclusión de estos bloques y realizar un programa con lenguaje mixto, es el caso del lenguaje LADDER.

En primer lugar, nos situamos en la línea del programa donde queremos introducir el bloque funcional. A continuación, seleccionaremos el icono denominado "Box".

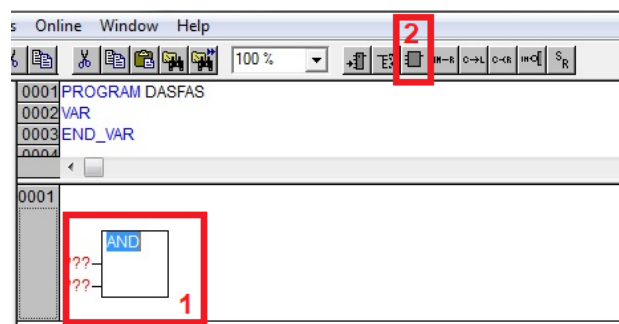


Figura 75 - CODESYS. Programación en FBD 3

Una vez introducida la caja nos situamos en la parte interior, donde se encuentra el nombre de la caja y apretamos “F2”. Se nos abrirá un cuadro de dialogo con todas las posibilidades existentes. Las diferentes funciones están ordenadas por tipos. En esta ventana podremos seleccionar bloques de librerías instaladas, funciones simples, bloques personalizados, o bien, llamar directamente una POU de nuestro programa.

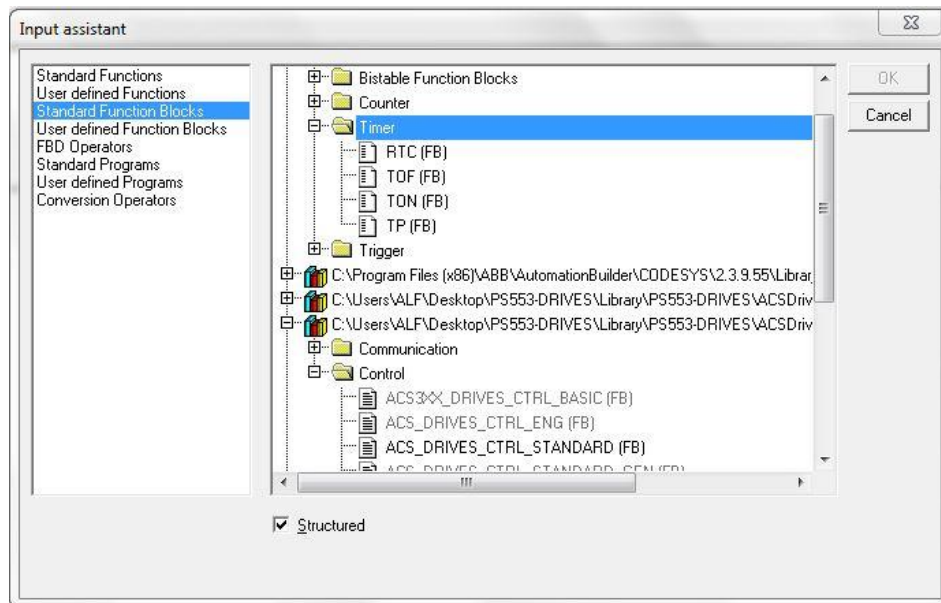


Figura 76 - CODESYS. Programación en FBD 4

Una vez tenemos el bloque funcional en la POU, se deberán introducir las variables de entrada o de salida. Para la introducción de las variables, únicamente, deberemos escribir su nombre en el enlace deseado, o bien, realizar el proceso explicado en el apartado 7.3.3. También existe la posibilidad de introducir una entrada o salida directamente a un nuevo bloque. Para ello, realizaremos el mismo proceso, con la particularidad que inicialmente deberemos situar el ratón en la E/S que deseamos situar el bloque y no en la fila del programa, como se ha realizado anteriormente.

Bloques funcionales de la librería PS553

En la programación del sistema de control del presente proyecto, se han utilizado los bloques funcionales de la librería “PS553-DRIVES”. Los bloques funcionales de esta librería se pueden utilizar debido a que los dos dispositivos del proyecto son ABB. En caso contrario, se

deberían usar otras librerías, o en su defecto, si el aplicativo es a través de MODBUS programar la palabra de control.

A continuación, se presentarán los módulos empleados en la programación de este proyecto, así como su posible parametrización para un correcto funcionamiento. Estos módulos se componen de entradas y salidas. Para un funcionamiento correcto, el modulo debe tener configuradas todas las entradas, en cambio, las salidas son informativas y pueden no ser configuradas o leídas. Las variables DRIVE_DATA deben ser la misma para todos los módulos de control de un mismo variador.

- **ACS_COM_MOD_RTU**

Este bloque funcional establece la comunicación entre el autómatas programable y el convertidor, a través del protocolo de comunicación Modbus RTU.

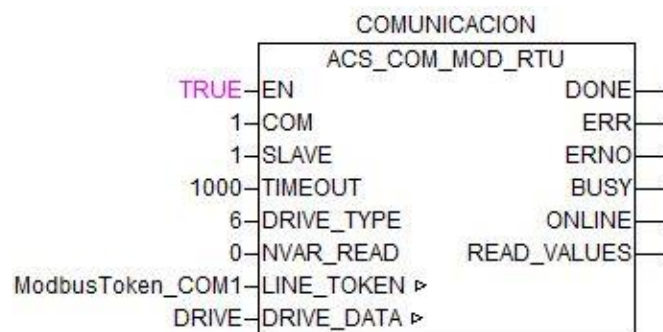


Figura 77 - CODESYS. Programación en FBD 5

Los variables configurables del módulo son las siguientes:

Variable	Tipo	Descripción
EN	BOOL	Activación del bloque. Debe proporcionar una señal TRUE constante
COM	BYTE	Puerto de comunicación Modbus 1 – COM1 2 – COM2
SLAVE	BYTE	Dirección de esclavo del convertidor
TIMEOUT	WORD	Tiempo de espera del telegrama (ms)
DRIVE_TYPE	INT	Tipo de variador: ACS550 = 6
NVAR_READ	BYTE	Numero de variables a leer.

		Corresponden a parámetros de comunicación.
LINE_TOKEN	ACS_MOD_TOKEN_TYPE	Variable de referencia del convertidor para intercambiar el testigo Modbus con otros bloques de función COM_MOD_RTU.
DRIVE_DATA	ACS_DRIVE_DATA_TYPE	Variable de referencia del convertidor para el intercambio de datos con todos los bloques de función relacionados con esta unidad.
DONE	BOOL	Salida. Ejecución terminada si estado es TRUE
ERR	BOOL	Salida. Se ha producido error estado igual a TRUE.
ERNO	WORD	Salida. Número del error producido
BUSY	BOOL	Salida. Control de comunicación/lectura
ONLINE	BOOL	Salida. Conexión realizada con éxito
READ_VALUES	ARRAY	Salida. Conjunto de valores leídos

Tabla 9 – Configuración E/S del bloque funcional ACS_COM_MOD_RTU

• ACS_DRIVES_CTRL_STANDARD

Este bloque funcional permite un control básico del variador de frecuencia. Así como la lectura del estado del convertidor.

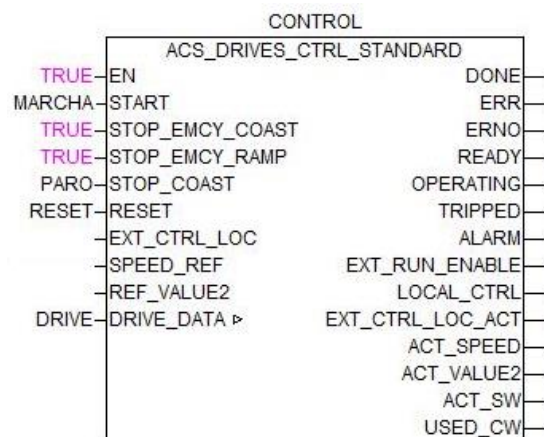


Figura 78 - CODESYS. Programación en FBD 6

Los variables configurables del módulo son las siguientes:

Variable	Tipo	Descripción
EN	BOOL	Activación del bloque. Debe proporcionar una señal TRUE constante
START	BOOL	Marcha y paro a partir de la rampa
STOP_EMCY_COAST	BOOL	Paro libre de emergencia
STOP_EMCY_RAMP	BOOL	Paro en rampa de emergencia
STOP_COAST	BOOL	Paro libre
RESET	BOOL	Restaurar convertidor y salida ERNO
EXT_CTRL_LOC	BOOL	Selección de control local
SPEED REF	REAL	Referencia de velocidad
REF_VALUE2	REAL	Referencia de par
DRIVE_DATA	ACS_DRIVE_DATA_TYPE	Variable de referencia del convertidor para el intercambio de datos con todos los bloques de función relacionados con esta unidad.
DONE	BOOL	Salida. Ejecución terminada si estado es TRUE
ERR	BOOL	Salida. Se ha producido error estado igual a TRUE.
ERNO	WORD	Salida. Número del error producido
READY	BOOL	Salida. Convertidor listo para la marcha
OPERATING	BOOL	Salida. El convertidor se controla mediante este bloque funcional
TRIPPED	BOOL	Salida. Disparo del convertidor
ALARM	BOOL	Salida. El convertidor presenta una alarma
EXT_RUN_ENABLE	BOOL	Salida.
LOCAL_CTRL	BOOL	Salida. Convertidor en modo local
EXT_CTRL_LOC_ACT	BOOL	Salida. Convertidor control en EXT2
ACT_SPEED	REAL	Salida. Velocidad actual

ACT_VALUE2	REAL	Salida. Par actual
ACT_SW	WORD	Salida. Palabra de estado leída
USED_CW	WORD	Salida. Palabra de estado aplicada

Tabla 10 - Configuración E/S del bloque funcional ACS_DRIVES_CTRL_STANDARD

• ACS_MOD_READ_N_PRM

Este bloque permite la lectura de uno o varios parámetros del variador de frecuencia.

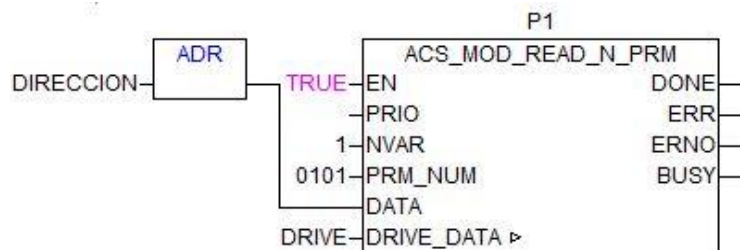


Figura 79 - CODESYS. Programación en FBD 7

Los variables configurables del módulo son las siguientes:

Variable	Tipo	Descripción
EN	BOOL	Activación del bloque funcional
PRIO	BOOL	Reservado. Sin uso
NVAR	UINT	Numero de variables a leer
PRN_NUM	UINT	Parámetro del convertidor a leer. En caso de leer varias, será la inicial. Los parámetros deben ser consecutivos.
DATA	DWORD	Almacenamiento del dato leído
DRIVE_DATA	ACS_DRUVE_DATA_TYPE	Variable de referencia del convertidor para el intercambio de datos con todos los bloques de función relacionados con esta unidad.
DONE	BOOL	Salida. Mensaje "Ready" (listo)
ERR	BOOL	Salida. Error del bloque
ERNO	WORD	Salida. Numero de error
BUSY	BOOL	Salida. Se está realizando el trabajo de lectura

Tabla 11 - Configuración E/S del bloque funcional ACS_MOD_READ_N_PRM

En la variable DATA debe intercalarse la función de un puntero (ADR). Los punteros almacenan las direcciones de variables, programas, bloques de funciones, métodos y funciones mientras se ejecuta un programa de aplicación. Un puntero señala uno de los objetos mencionados o una variable con cualquier tipo de datos. Este operador es una extensión de la norma IEC 61131-3. Para insertar el puntero, seleccionaremos un bloque funcional con el operando ADR.

- **ACS_MOD_WRITE_N_PRM**

Este bloque permite la modificación de uno o varios parámetros del variador de frecuencia.

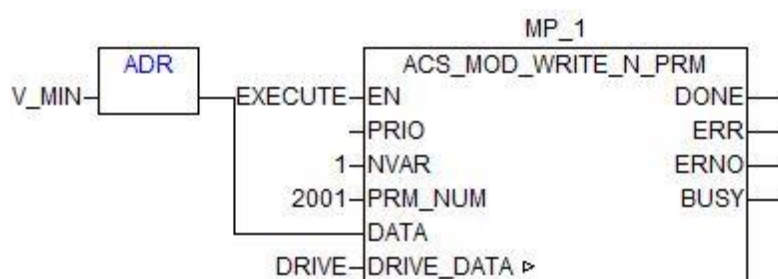


Figura 80 - CODESYS. Programación en FBD 8

Los variables configurables del módulo son las siguientes:

Variable	Tipo	Descripción
EN	BOOL	Activación del bloque funcional
PRIO	BOOL	Reservado. Sin uso
NVAR	UINT	Numero de variables a leer
PRN_NUM	UINT	Parámetro del convertidor a leer. En caso de leer varias, será la inicial. Los parámetros deben ser consecutivos.
DATA	DWORD	Dato que debe ser modificado
DRIVE_DATA	ACS_DRUVE_DATA_TYPE	Variable de referencia del convertidor para el intercambio de datos con todos los bloques de función relacionados con esta unidad.
DONE	BOOL	Salida. Mensaje "Ready" (listo)
ERR	BOOL	Salida. Error del bloque
ERNO	WORD	Salida. Numero de error
BUSY	BOOL	Salida. Se está realizando el trabajo de lectura

Tabla 12 - Configuración E/S del bloque funcional ACS_MOD_WRITE_N_PRM

En la variable DATA debe intercalarse la función de un puntero (ADR), como en el módulo anterior.

Personalización de un bloque funcional

La creación de un bloque funcional personalizado se puede realizar con dos objetivos: la utilización dentro del programa o la utilización en diversos programas. Cuando realizamos un bloque funcional personalizado para la utilización dentro de un programa se busca encontrar una solución programada a una tarea que no se puede realizar con los bloques ya existentes. En cambio, cuando realizamos un bloque funcional para la utilización en diversos programas es porque hemos detectado una tarea repetitiva que debemos solventar en ocasiones o proyectos distintos.

Para crear un bloque funcional y utilizarlo en distintos programas, debemos dirigirnos al árbol de POU's y seleccionar añadir nuevo objeto, como se ha realizado en la creación de una nueva POU.

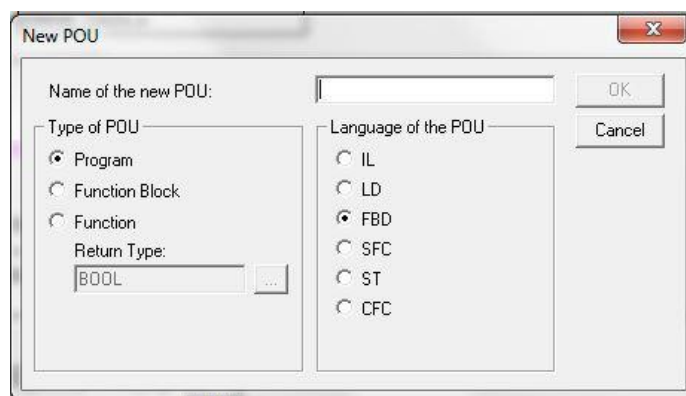


Figura 81 - CODESYS. Programación en FBD 9

La diferencia es que en lugar de seleccionar “Program”, seleccionaremos “Function Block”. Se nos abrirá una nueva POU, donde deberemos programar el bloque funcional. Una vez guardada, podremos seleccionar ese bloque funcional en la lista de “Input Assistant”, como se ha realizado anteriormente para añadir un bloque funcional.

Para realizar un bloque funcional, que solo utilizaremos en un mismo proyecto, realizaremos una nueva POU. En esta nueva POU, se programará la tarea que deba desarrollar el bloque funcional.

A continuació, introducirem el bloque funcional en la part del programa que se requiera. Al obrir el “Input Assistant” seleccionarem la opció “Standard Program”, on apareixeran les distintes POU’s del programa.

En la figura precedent, se puede observar un bloque funcional que llama a una POU del programa de control de este proyecto.

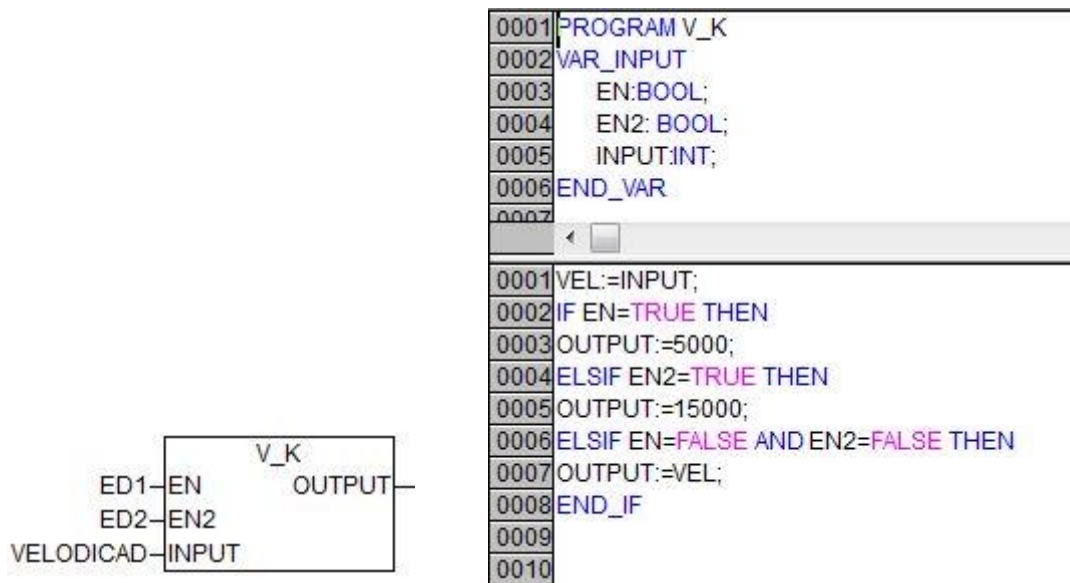


Figura 82 - CODESYS. Programación en FBD 10

7.3.5. Visualizaciones

CODESYS permite el desarrollo de pequeñas interfaces HMI para la comprensión, control y simulación de los procesos. Esto permite un mayor entendimiento del programa y una facilidad a la hora de testear el correcto funcionamiento de la aplicación.

Estas visualizaciones no están destinadas a ser descargadas en un dispositivo de control, sino a ser una herramienta de desarrollo que complemente el editor del programa.

Creación de una visualización

Para la creación de una visualización debemos dirigirnos a la pestaña de “Visualizations”. En la nueva ventana, clicaremos con el botón izquierdo del ratón y seleccionaremos “Add Object”.

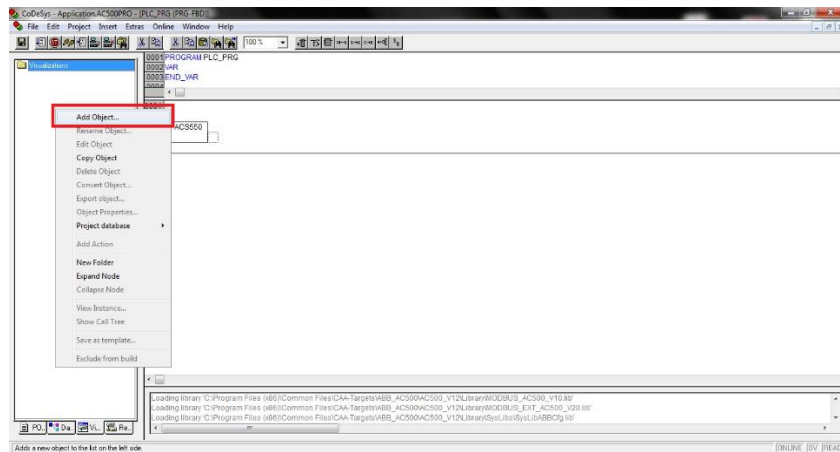


Figura 83 – CODESYS. Creación de visualizaciones 1

A continuación, en la nueva ventana se introducirá el nombre de la nueva visualización.

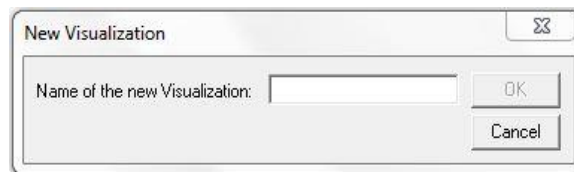


Figura 84 - CODESYS. Creación de visualizaciones 2

Finalmente, se nos abrirá el editor de visualizaciones con una página en blanco. Es el espacio donde podremos crear la visualización. En la parte superior, tenemos las herramientas y objetos que se pueden incluir en una visualización.

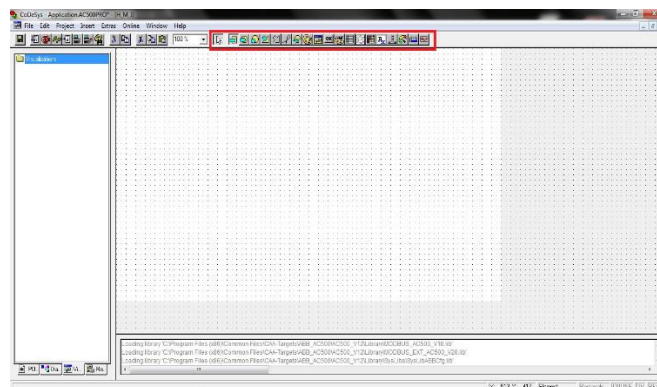


Figura 85 - CODESYS. Creación de visualizaciones 3

CODESYS permite la inclusión de elementos animados o algunos formatos de objetos como JAVA o WMF. Estos elementos no serán empleados en el presente proyecto, ya que son necesarias bibliotecas de fabricantes que no son de libre acceso. A continuación, se

establecerán las herramientas básicas para la creación de objetos desde su inicio, con las herramientas dispuestas por CODESYS.

Editor de visualizaciones

Las herramientas de CODESYS se extienden en la barra de tareas de la parte superior del editor. Permite la creación de elementos, líneas o poliformas, así como botones o objetos SCADA. Al sobreponer el ratón sobre cada uno de los iconos, CODESYS nos indicara el tipo de elemento para añadir.



Figura 86 - CODESYS. Creación de visualizaciones 4

- **Inserción de objetos:** la inserción de objetos se realiza del mismo modo sin importar su naturaleza. Una vez se ha insertado, si se realiza doble clic sobre el objeto se accede a la ventana de configuración.

Para insertar un nuevo objeto, seleccionaremos el icono del objeto que se desea insertar. Nos dirigimos al editor y realizamos un clic, en el punto de inicio del objeto. Sin dejar de presionar el botón del ratón, trazamos la diagonal, para dar el tamaño requerido. Una vez tenemos la forma podemos configurar el elemento. Una misma forma, según se configure puede realizar acciones diferentes.

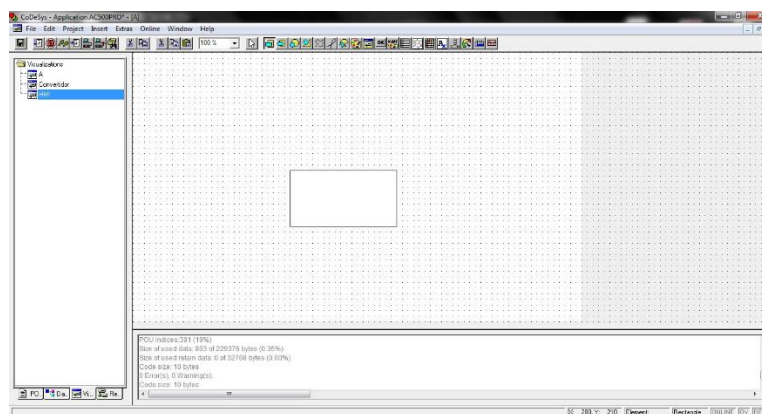


Figura 87 - CODESYS. Creación de visualizaciones 5

- **Botones:** se pueden insertar dos tipos de botones, prediseñados por CODESYS o realizar una autoforma con las funciones de botón.

Si realizamos una autoforma con las herramientas de CODESYS. Una vez tenemos la forma deseada, haciendo doble clic entramos a la ventana de configuración. En la categoría de “Text”, podremos indicar una etiqueta que se visualizará en el botón. La topología de letra es modificable.

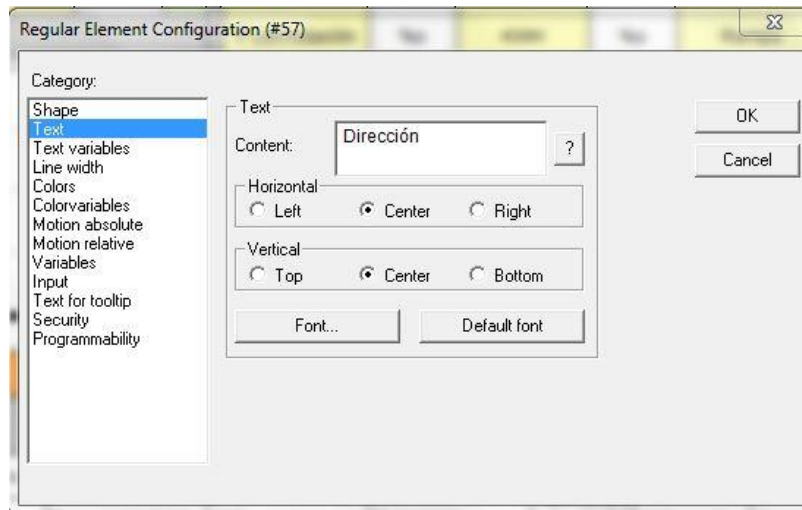


Figura 88 - CODESYS. Creación de visualizaciones 6

Para indicar la variable asociada, nos dirigimos a la categoría “Input”, y tenemos dos posibilidades. En “Toggle variable” el botón funcionará como un interruptor, en cambio si se selecciona “Tap variable”, el botón hará la función de un pulsador.

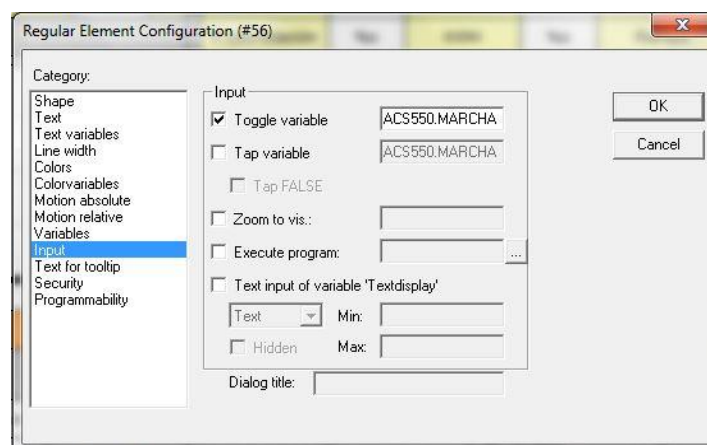



Figura 89 - CODESYS. Creación de visualizaciones 7

En las otras categorías del menú, podremos configurar parámetros de diseño como colores u otras funcionalidades.

Para insertar un botón prediseñado seleccionaremos el icono , después únicamente se habrá de introducir la variable asociada, del mismo modo anterior. Al ser un botón prediseñado muchas de las funcionalidades anteriores del menú no estarán presentes, ya que vienen definidas por defecto.

- **Elementos de señalización:** para mostrar el cambio de estado de una variable booleana, es posible insertar un objeto que cambie de color para cada estado. Por ejemplo, se puede insertar una circunferencia a modo de piloto de señalización. Una vez tenemos definida la forma. Entramos en el menú de configuración. En la categoría “Colors” indicamos dos colores distintos, en estado normal y en modo alarma.

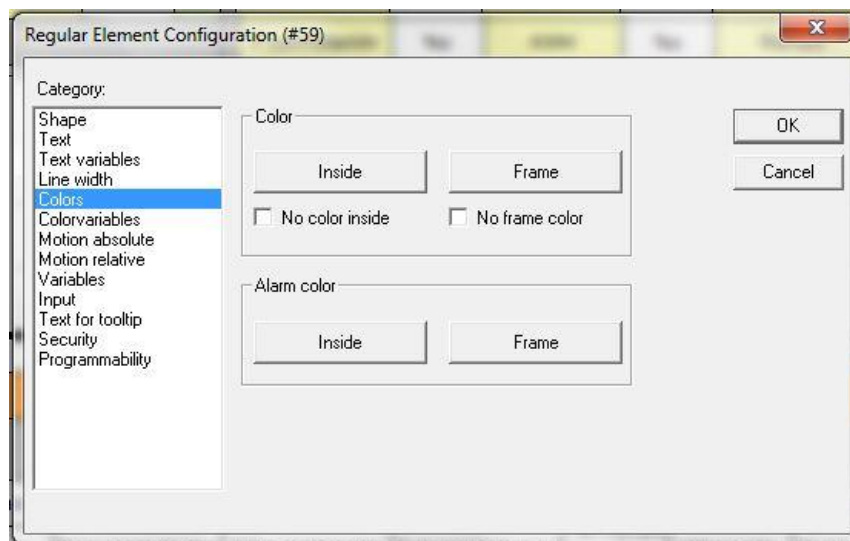


Figura 90 - CODESYS. Creación de visualizaciones 8

A continuación, se debe asociar el cambio de color a una variable. Para ello, accedemos a la categoría “Variables”. En esta categoría se presentan distintas acciones, para el cambio el estado de una variable. En este caso, se seleccionará “Change color”, y se escribe la variable que queremos asociar. También es posible, presionar “F2” para acceder a la lista de variables.

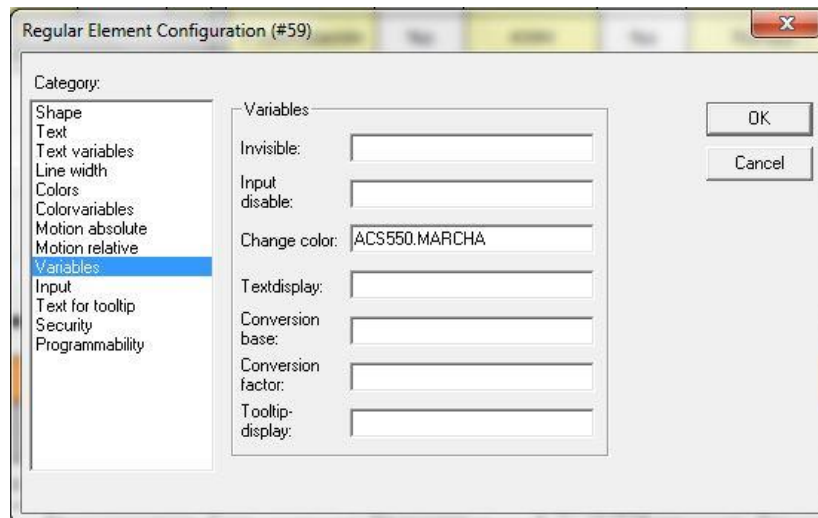



Figura 91 - CODESYS. Creación de visualizaciones 9

- Histogramas:** para la creación de un histograma seleccionamos el icono  de la barra de tareas. A continuación, indicamos en el editor el tamaño y ubicación del objeto. Hacemos doble clic en el objeto, recientemente creado, y se abre la ventana de configuración. En la ventana de configuración, tenemos 4 opciones. "Security" nos permite restringir el acceso al elemento. "Text for tooltip" permite añadir un comentario. "Colors" permite la modificación y configuración de los colores de la visualización. Son modificables los fondos, ejes o curvas. En "trend" se deben configurar los ejes y su ubicación, también se pueden parametrizar valores de la curva en "Curve configuration".

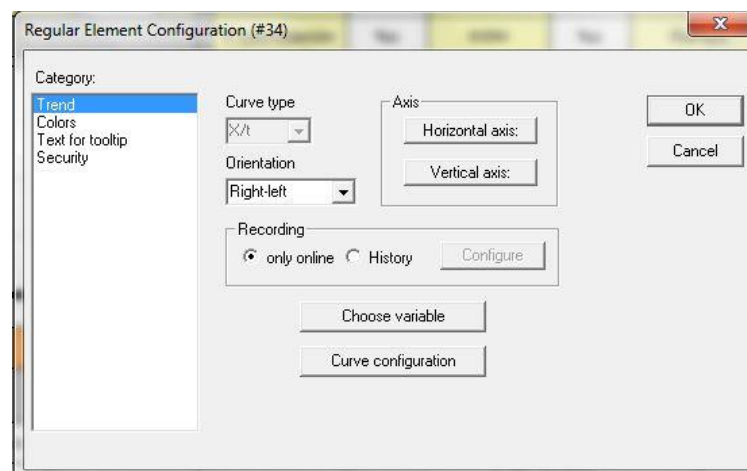


Figura 92 - CODESYS. Creación de visualizaciones 10

Para la selección de la variable que queremos visualizar, clicamos en “Choose variable”. En la nueva ventana, clicamos en “Add”. Aquí se dispondrán todas las variables del sistema y podremos añadir la variable deseada. Existe la posibilidad de añadir múltiples variables, únicamente, debemos repetir el proceso.

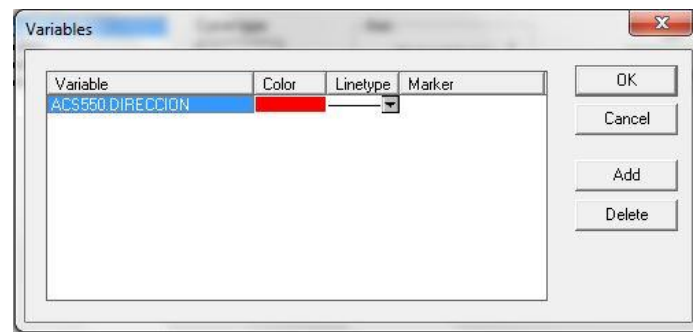


Figura 93 - CODESYS. Creación de visualizaciones 11

- **Visualización de parámetros:** para mostrar el valor de un parámetro o variable, insertaremos la forma deseada. Accedemos al menú de configuración, y en la categoría “Text” escribiremos “%s”. A continuación, en la categoría “Variables” en la opción “Textdisplay” indicaremos la variable que se quiera mostrar. Cuando iniciemos, el modo simulación u online, el valor de la variable aparecerá en la forma que hemos creado. Es posible que la lectura de algunos parámetros no se realice en el formato o las unidades que se desea mostrar. En este caso en la opción “Textdisplay”, insertaremos la variable dentro de una formula.

Parametros nominales del motor					
Tensión	%s	V	Potencia	%s	W

Figura 94 - CODESYS. Creación de visualizaciones 12

- **Modificación de parámetros:** para la modificación de un parámetro, debemos asociar una variable numérica, por ejemplo, un entero. En primer lugar, insertaremos la forma requerida y a continuación entraremos en el menú de configuración. Accedemos a la categoría de “Input” y seleccionamos la opción “Test input of variable ‘Text display’”. En esta opción seleccionamos el teclado

que queremos que se nos muestre, que dependerá del tipo de variable a modificar. A continuación, seleccionamos los límites de dicha variable.

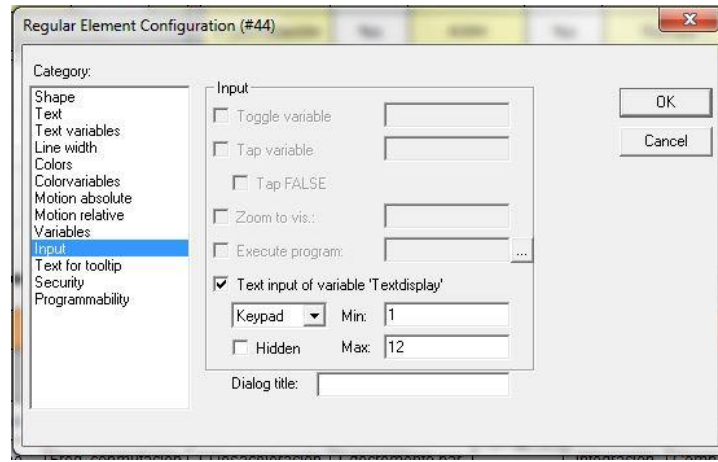


Figura 95 - CODESYS. Creación de visualizaciones 13

Una vez, iniciemos el modo online o modo simulación al presionar sobre la forma que hemos insertado, se nos abrirá el teclado para modificar el parámetro asociado.




Figura 96 - CODESYS. Creación de visualizaciones 14

Estas son algunas de las funcionalidades de uso más común. La mayoría de opciones que se han mostrado en este apartado son combinables, es decir, existe la posibilidad de diseñar un bloque que muestre la variable y, al mismo tiempo, pueda modificarla o un botón que cambie de color al presionarlo. Asimismo, estas formas pueden no estar asociadas a una variable y formar parte del diseño o mostrar un texto a modo de indicación. Existen otros tipos de funcionalidades animadas, como el movimiento de una autoforma a través de un recorrido prediseñado. Esto permite la realización de visualizaciones animadas.

Visualizaciones estándar librería PS553

La librería PS553-DRIVES dispone de visualizaciones asociadas a los diferentes bloques funcionales. En ella se mostrarán las distintas variables del módulo, así como las variables adicionales que puedan ser leídas del convertidor. Es una forma fácil de comprobar el correcto funcionamiento de estos módulos, teniendo toda la información completa y fácilmente accesible.

Para insertar un módulo de estas características, seleccionaremos el botón  de la barra de herramientas. A continuación, en el área de trabajo se le proporcionará la forma deseada.

Automáticamente, se abrirá un nuevo cuadro de dialogo donde aparecerán las distintas visualizaciones disponibles. Seleccionaremos, la visualización asociada al bloque funcional que deseamos mostrar.

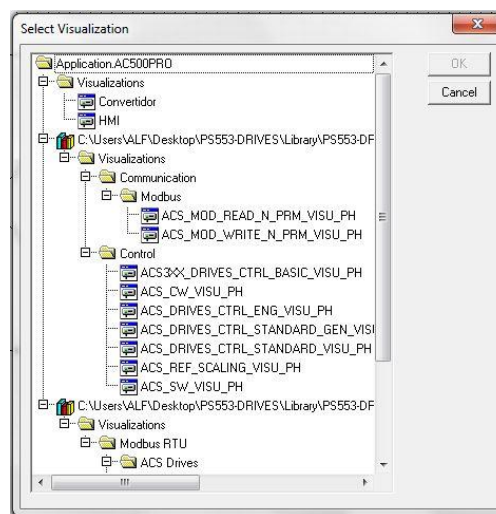


Figura 97 - CODESYS. Creación de visualizaciones 15

Para el desarrollo del presente proyecto, se utilizaron las visualizaciones que se muestran a continuación. Son solo un ejemplo de las visualizaciones disponibles en esta librería.

ACS_COM_MOD_RTU			
\$FB\$			
%S EN	DONE	%S	
%S COM	ERR	%S	
%S SLAVE	ERNO	%S	
%S TIMEOUT	BUSY	%S	
%S DRIVE_TYPE	ONLINE	%S	
%S NVAR_READ	WriteErrCnt	%S	
	LastWriteErr	%S	
	ReadErrCnt	%S	
	LastReadErr	%S	
	MSW	%S	
%S MCW	ActValue1	%S	
%S RefValue1	ActValue2	%S	
%S RefValue2			
READ_VALUES - for embeded modbus			
%S DAA_IN 1	DAA_IN 13	%S	
%S DAA_IN 2	DAA_IN 14	%S	
%S DAA_IN 3	DAA_IN 15	%S	
%S DAA_IN 4	DAA_IN 16	%S	
%S DAA_IN 5	DAA_IN 17	%S	
%S DAA_IN 6	DAA_IN 18	%S	
%S DAA_IN 7	DAA_IN 19	%S	
%S DAA_IN 8	DAA_IN 20	%S	
%S DAA_IN 9	DAA_IN 21	%S	
%S DAA_IN 10	DAA_IN 22	%S	
%S DAA_IN 11	DAA_IN 23	%S	
%S DAA_IN 12	DAA_IN 24	%S	

ACS_DRIVES_CTRL_STANDARD_GEN			
\$FB\$			
%S EN	DONE	%S	
%S START	ERR	%S	
%S EMCY_COAS	ERNO	%S	
%S EMCY_RAMP	READY	%S	
%S STOP_COAS	OPERATING	%S	
%S RESET	TRIPPED	%S	
	ALARM	%S	
	EXT_RUN_EN	%S	
	LOCAL_CTRL	%S	
%S EXT_CTRL_L	EXT_CTRL_L	%S	
%S SW	CW	%S	
MESSAGE		%S	

Figura 98 - CODESYS. Creación de visualizaciones 16

7.3.6. Modo online y modo simulación

Modo simulación

CODESYS dispone de un modo simulación para la comprobación y el testeo de la aplicación. El modo simulación permite el control desde el editor del programa o desde las visualizaciones. En este modo se dispone el área de trabajo para modificar y observar el comportamiento de las variables.

Para iniciar el modo simulación nos dirigimos al menú "Online" y seleccionamos "Simulation mode". A continuación, en el mismo menú clicamos en "Login". En este instante, el área de trabajo, ya se modifica al modo simulación y podemos realizar los cambios iniciales que sean necesarios.

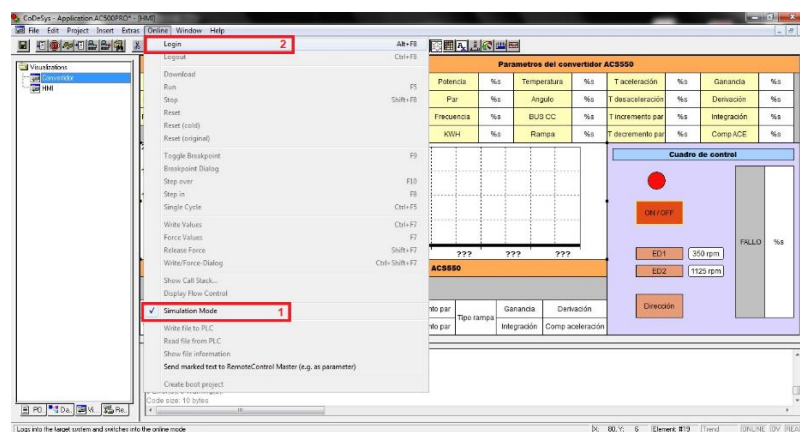


Figura 99 - CODESYS. Modo simulación 1

Una vez se tienen los cambios iniciales, en el menú “Online” seleccionamos “Run”. A partir de este momento el programa ya está corriendo y podremos observar los cambios en el estado de las variables.

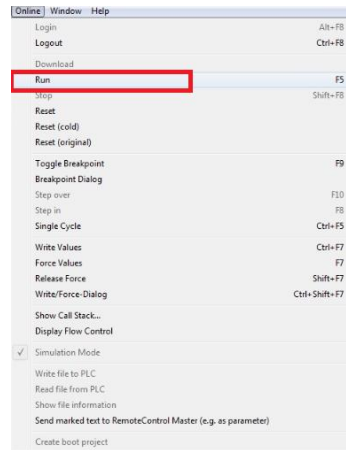


Figura 100 - CODESYS. Modo simulación 2

Si nos encontramos en el modo simulación desde la interfaz gráfica de “Visualizations”, podremos modificar directamente las variables desde los objetos creados. El modo simulación también funciona desde el editor del programa. El proceso es el mismo. CODESYS permite alternar entre el editor y las visualizaciones en el modo simulación, de manera que se pueden modificar variables desde distintos puntos.

En el editor podremos modificar las variables desde el programa, o bien, en la ventana de declaración. Para ello, debemos hacer doble clic en la variable que se desea modificar. En caso de ser booleana, el editor ira cambiando de estado dicha variable. Si es una variable alfanumérica, se nos abrirá un teclado para poder realizar los cambios.

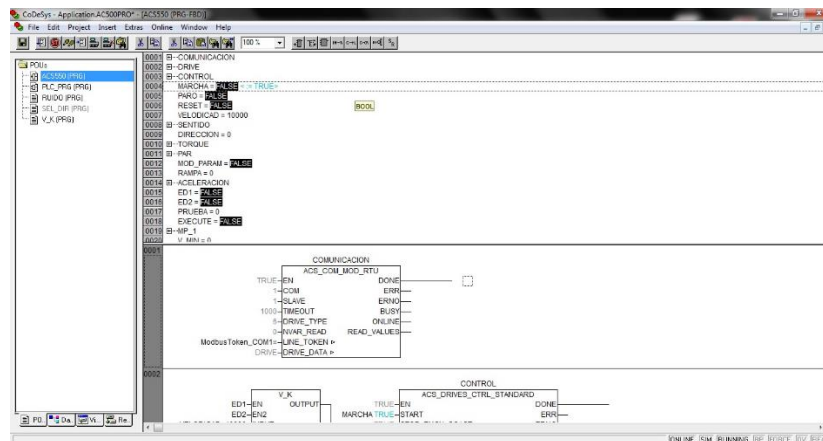


Figura 101 - CODESYS. Modo simulación 3

A diferencia del modo de simulación en las visualizaciones, en el editor si queremos que una modificación del estado de una variable cambie, deberemos forzar el valor. Se puede realizar presionando “F7”, o bien desde el menú “Online” donde deberemos clicar sobre “Force Values”.

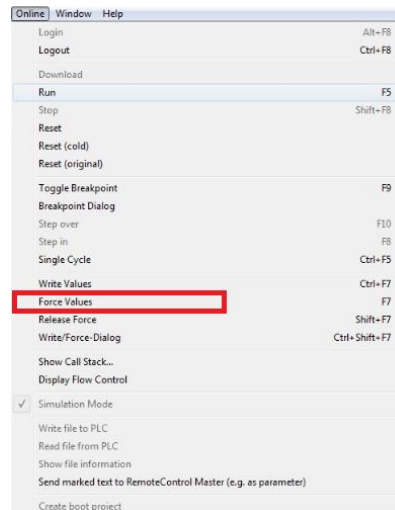


Figura 102 - CODESYS. Modo simulación 4

Una vez hemos forzado el valor, la variable se modifica y podemos observar que dicha variable se subrayará de color azul. Esto indicará que la modificación se ha producido.



Figura 103 - CODESYS. Modo simulación 5

Modo online

El modo online permite iniciar el programa en el autómata. Para iniciarlo deberemos acceder al menú “Online” y seleccionar “Login”. Es necesario asegurarse que la casilla de “Simulation Mode” del menú “Online” no está activada.

El software compilará el programa automáticamente, si detecta que el programa no coincide en su totalidad con el que está guardado en el autómata, indicará si se quiere guardar el nuevo proyecto. En caso de que se haya modificado algún aspecto del programa y el

software no lo detectara, se puede guardar en la memoria flash del autómatas seleccionando “Create boot Project” en el menú “Online”.

Para iniciar el sistema, al igual que en el modo simulación, se debe seleccionar “Run”, en el menú “Online”. En este modo también existe la posibilidad de forzar los valores de las variables, el proceso es el mismo que el indicado en el modo simulación.

El inicio y descarga en el autómatas del programa, también se puede realizar desde el software “Automation Builder” (Véase apartado 7.2).

7.4.Panel Builder 600

El software para la programación del sistema HMI es Panel Builder 600. Es un software de ABB, incluido en la instalación de Automation Builder. Está basado en tecnología XML y permite la creación de objetos gráficos fácilmente.

Los objetos visuales son SVG (Scalable Vector Graphics). Estos se caracterizan por tener un formato bidimensional y pueden ser animados o estáticos. Son creaciones, normalmente, de código abierto, y en consecuencia, ofrecen un alto grado de flexibilidad y personalización.

El software Panel Builder admite la integración de gran parte de las comunicaciones existentes, entre las que destaca Modbus RTU, protocolo empleado en este proyecto.

Las diversas funcionalidades lo convierten en una potente herramienta de automatización. Permite la adquisición de datos, sistemas multilenguaje, históricos de alarma, capas múltiples e incluso menús rotativos. Todo ello, con la posibilidad de desarrollarlo en colores de 64k y de una forma muy intuitiva.

Para la integración de Panel Builder en un proyecto de Automation Builder, y tener la posibilidad de programar todo el proyecto desde un mismo entorno es necesario una cuenta Avanzada. Si, únicamente, se tiene en posesión la cuenta gratuita Básica, el proyecto se deberá programar por separado. (Para más información véase apartado 7.4.3)

7.4.1. Explicación del entorno

El software Panel Builder 600 mantiene un entorno semejante al de Automation Builder. El entorno del software se puede dividir en cuatro partes diferenciadas. Seguidamente, se especificarán las funciones y opciones que ofrecen cada una de los apartados. Algunas partes del menú desplegable, tienen botones que realizan la misma función el menú de inicio rápido. Se pueden identificar con facilidad, ya que los iconos del menú desplegable y del inicio rápido son idénticos.

En el cuadro 1 de la figura 105, tenemos la barra de menú y los iconos de inicio rápido. Es la parte de gestión del programa, y desde ella, accederemos a las diversas funcionalidades del software. A continuación, se detallan algunas de funcionalidades más importantes de los diferentes menús.

- File: el menú da acceso a la gestión de archivos. Permite crear un nuevo proyecto, abrirlo o guardarlo.
- Edit: se dispone de las herramientas de edición, como deshacer, cortar o pegar.
- Run: es el menú de conexión con el panel. Permite la conexión y gestión del programa respecto al dispositivo HMI. También permite acceder al modo simulación. Algunas de sus funcionalidades también están disponibles en el menú de inicio rápido.
- Format: proporciona herramientas para dar formato a los objetos insertados en el área de trabajo como tamaño, orientación o alineado. Estas tareas se pueden realizar de manera gráfica en el área de trabajo, o mediante el menú de propiedades de forma numérica.
- View: ofrece la gestión de visualización, tanto para el proyecto como para las diferentes barras de herramientas. También proporciona la posibilidad de añadir una cuadrícula para el diseño del panel.

- Windows: gestiona las ventanas del proyecto.
- Help: acceso al documento de ayuda del software. Proporciona información sobre como programar el panel y de los diferentes dispositivos compatibles con el software.

En el cuadro 2 de la figura 105, se tiene el árbol del proyecto. Aquí se mostrarán las diferentes páginas, diálogos o plantillas. Así como las herramientas de comunicación o funcionalidades del panel. Para acceder a ellas, se realiza un doble clic. La nueva pantalla se mostrará en el cuadro 3 de la figura 105.

En el cuadro 3 de la figura 105, se muestra el área de trabajo. Corresponde con las medidas del panel que se ha configurado. Es la hoja en blanco donde se diseña y configura el panel de control. En la parte superior, se muestran algunas herramientas básicas de trabajo, como el “Zoom” o la tipología de letra.

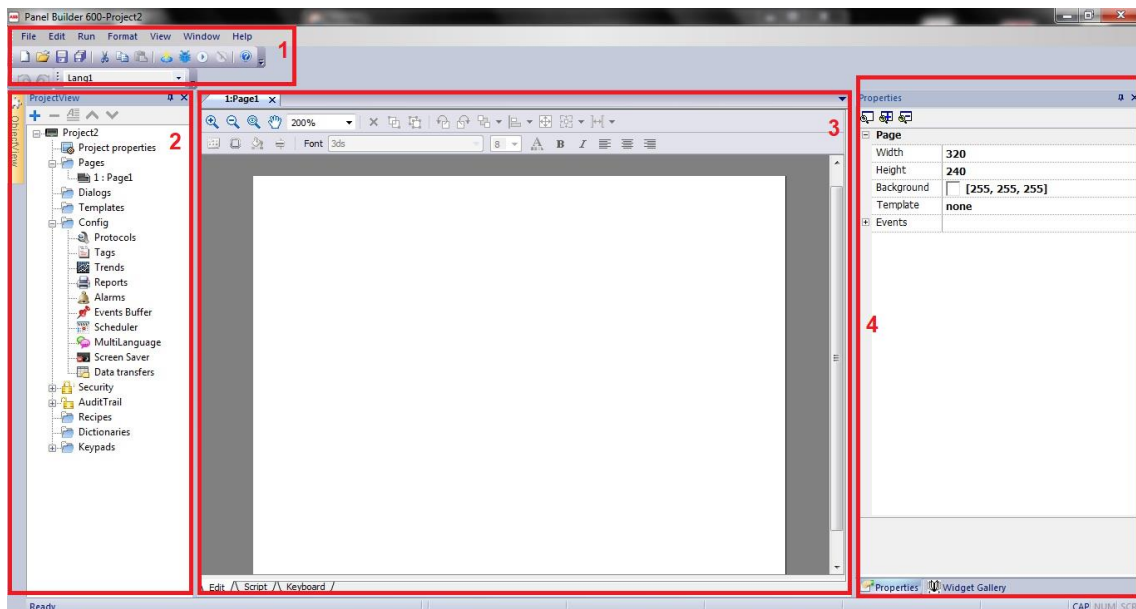


Figura 104 – Panel Builder. Entorno

En el cuadro 4 de la figura 105, disponemos de dos pestañas. En la pestaña “Propiedades” se muestran las características modificables del elemento, ya sea un elemento del árbol de proyecto o el objeto del área de trabajo que tenemos seleccionado. En la pestaña de “Widget

Gallery”, se disponen de los diferentes elementos gráficos que podemos agregar al diseño de la pantalla.

7.4.2. Creación de un proyecto nuevo

Para la creación de un nuevo proyecto, iniciamos el software Panel Builder 600. Una vez, se tiene el programa abierto nos dirigimos al icono de “New”, como se observa en la figura 105. También existe la posibilidad de iniciar el proyecto a través del menú, para ello se debe acceder a la ruta “File” y posteriormente a la opción “New”.

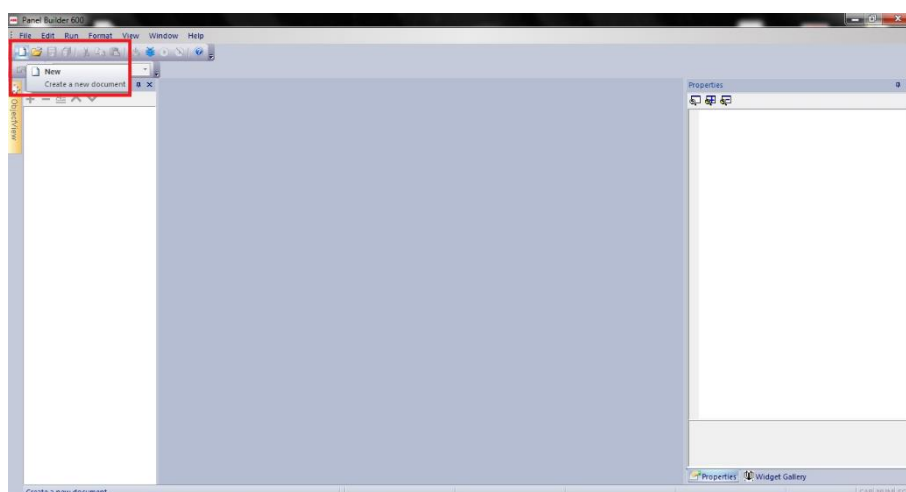


Figura 105 – Panel Builder. Creación nuevo proyecto 1

Se abrirá un primer cuadro de dialogo, en el cual, debemos introducir el nombre del proyecto y la ubicación donde se guardarán los archivos. En este momento, podemos optar por finalizar el proceso, y configurar el panel con posterioridad, o bien, clicar a “Next” para configurar el panel. Si configuramos el panel, el nuevo documento se abrirá con las dimensiones correctas de nuestra pantalla. Es un hecho importante, ya que, si este proceso se realiza con posterioridad, podemos tener problemas de resolución por la modificación del tamaño los elementos gráficos.

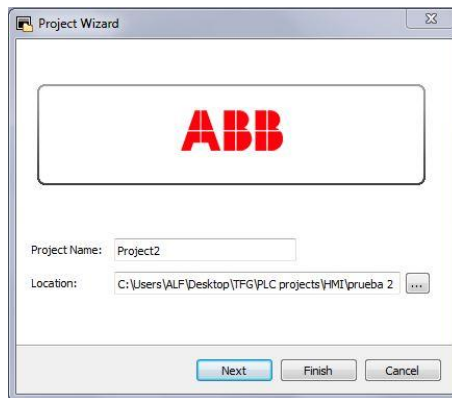


Figura 106 - Panel Builder. Creación nuevo proyecto 2

Si hemos seleccionado continuar, se nos abrirá un segundo cuadro de dialogo. En esta ventana, seleccionaremos el panel, en nuestro caso CP630 (véase apartado 6.4.3). Al seleccionar el panel, el software configura el tamaño del área de trabajo automáticamente. En caso de no disponer de un panel ABB, deberemos finalizar el proceso en la ventana anterior y configurar las dimensiones del área de trabajo de manera manual. Finalmente se selecciona la orientación del panel y clicamos en “Finish”.

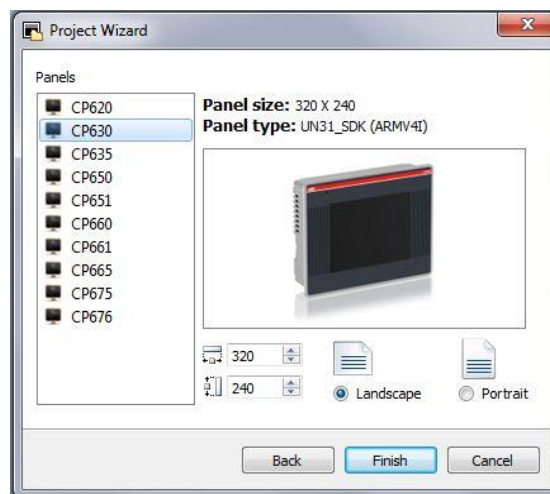


Figura 107 - Panel Builder. Creación nuevo proyecto 3

Una vez finalizado el proceso, se nos abrirá el software en la página principal, con el área de trabajo en blanco. Una vez finalizada la creación del proyecto, si necesitamos modificar el panel debemos dirigirnos a “Project View” y “Project Propierties”. El tipo de panel, únicamente, modifica las dimensiones de la pantalla y, es en esta sección, donde podremos modificar el tamaño del área de trabajo. Este proceso es el mismo que deberemos seguir, si nuestro panel no es del fabricante ABB.

7.4.3. Importación de variables

El software Panel Builder permite la creación de variables para su uso y ejecución en el programa del panel del control. La creación de variables se realiza desde la pestaña “Tags”. Sin embargo, en el presente proyecto se utilizarán variables compartidas que ya han sido creadas con anterioridad en los softwares Automation Builder y CODESYS.

Las variables empleadas en el programa del PAC, deben ser exportadas al nuevo software. Estas variables deben indicar implícitamente el tipo de variable, pero también, la dirección de la memoria donde se guardan. Esto permitirá que, a través del bus de campo, la lectura o modificación del de una dirección, y por tanto, el cambio de valor de la variable.

Para la exportación de las variables, debemos dirigirnos a la pestaña “Resources” del software CODESYS. En el apartado de variables globales, seleccionamos el archivo de variables globales donde se han declarado las variables. Clicamos con el botón izquierdo y seleccionamos “Export Object”.

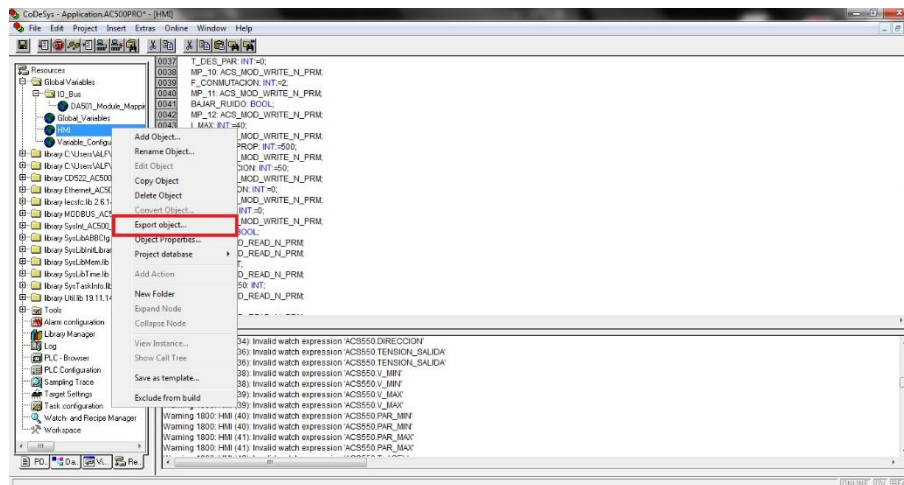


Figura 108 - Panel Builder. Importación de variables 1

En la nueva ventana de dialogo, deberemos guardar el archivo “.exp” en la ubicación deseada. Este archivo es el que debemos importar en el software Panel Builder. El formato de archivo y el proceso de exportación es diferente según el protocolo de comunicación empleado, por ejemplo, para comunicaciones “TCP” debe exportarse el archivo de símbolos “.sym”.

Una vez tenemos exportado el archivo, nos dirigimos a la pestaña “Tags” del árbol de proyecto. Nos aseguramos de tener seleccionado el protocolo de comunicación adecuado, en este caso, “ABB Modbus RTU” que ha configurado con anterioridad. A continuación, clicamos en el botón, inmediatamente, a la izquierda que es el de “Import Tags”

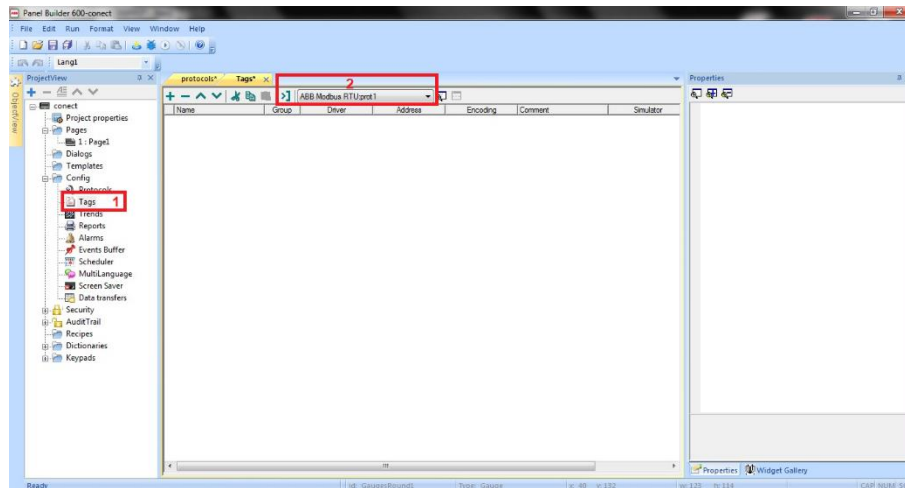


Figura 109 - Panel Builder. Importación de variables 2

En la ventana de dialogo se nos muestra el tipo de archivo y el protocolo a seleccionar. Seleccionamos “ABB Modbus RTU” y el archivo “.exp”.

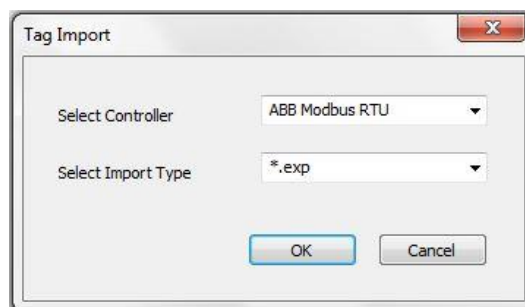


Figura 110 - Panel Builder. Importación de variables 3

A continuación, seleccionaremos el archivo en la ubicación donde ha sido guardado y aceptamos el proceso. En este momento, volveremos a la página inicial. Se puede observar que la ventana ha sido dividida en dos partes. En la parte inferior tenemos las variables declaradas en el archivo y en la parte superior las que han sido declaradas en Panel Builder. Para traspasar las variables al programa actual, seleccionamos las que sean necesarias y en la barra de tareas inferior seleccionamos “Import Tags”.

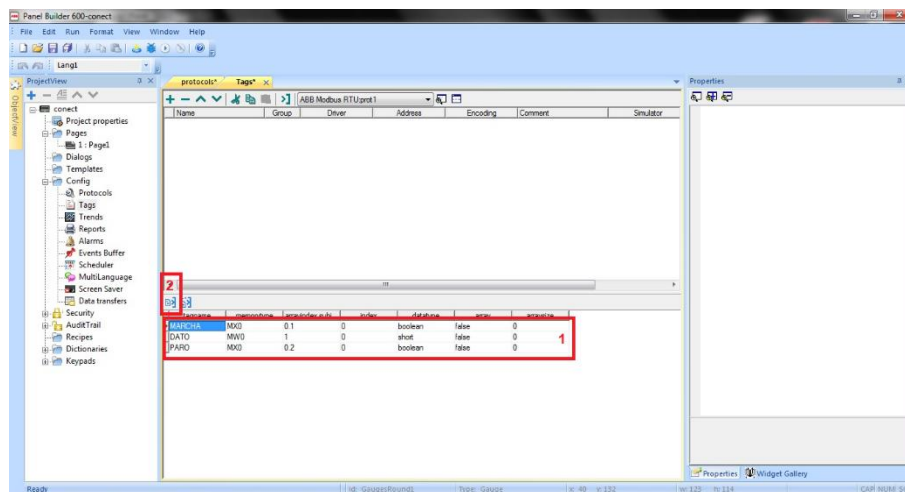


Figura 111 - Panel Builder. Importación de variables 4

Si el proceso se ha realizado correctamente, las variables seleccionadas se ubicarán automáticamente en la parte superior del programa. Estas variables, como se ha comentado con anterioridad, corresponden a las variables del programa y, por tanto, ya pueden ser utilizadas en el diseño HMI del panel.

Se puede observar que cada variable se separa por columnas, donde se muestra la información del tipo de variable, el tipo de memoria y la dirección. Todos estos valores son editables.

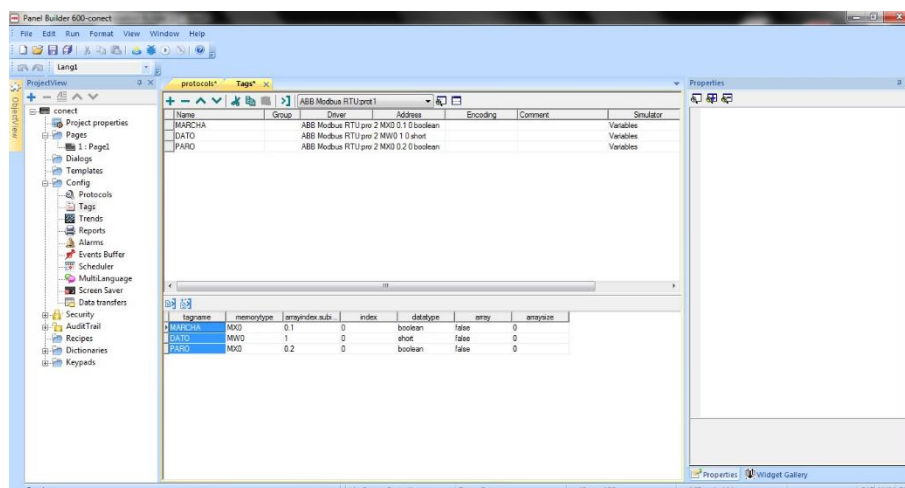


Figura 112 - Panel Builder. Importación de variables 5

7.4.4. Programación y diseño

El software Panel Builder es un entorno de programación visual, que permite la inserción de objetos estáticos o dinámicos que realizan una función concreta. Además, de las propiedades del objeto, se integra la posibilidad de la programación orientada a eventos. Esto quiere decir, que la actuación por parte del usuario sobre un objeto activará una funcionalidad concreta. Todo el desarrollo y programación del panel se realiza sin la necesidad de utilización de código.

Creación de páginas y diálogos

En un diseño HMI se dispone de diversas paginas que enlazan unas con otras, o bien, cuadros de dialogo que se abren al ejecutar una acción. Para crear nuevas paginas, se debe seleccionar el apartado “Pages” en el arbol de proyecto con el botón izquierdo del ratón y seleccionar “Insert New Page”. Consecutivamente, le asignaremos un nombre. Este nombre funcionará como el de una variable, es decir, cuando se crea un vínculo para acceder a dicha pagina, lo haremos asignando ese nombre a la acción.

La creación de cuadros de dialogo se realiza de la misma manera. Pero a diferencia del proceso anterior, clicaremos en “Dialogs” y seguidamente “Insert New Dialog Page”. En la configuración de los cuadros de dialogo, es importante seleccionar el tamaño adecuado en el cuadro de propiedades. Ya que, el cuadro puede quedarse fuera de la parte visual del panel de control e impedir realizar la acción correspondiente.

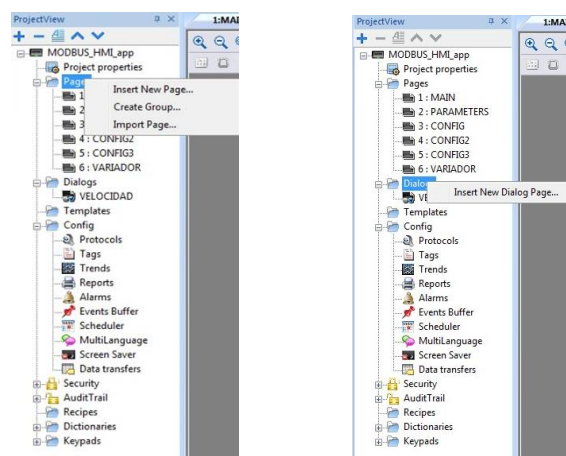


Figura 113 - Panel Builder. Programación 1

Objetos

En la galería de objetos, situada en la parte derecha del software, disponemos de una gran variedad de objetos destinados a una funcionalidad concreta. Los elementos disponibles son tipos de texto, botones, luces, elementos gráficos de automatización y herramientas avanzadas, entre otros. Estos elementos pueden complementarse con la instalación de nuevas librerías que aumentarían el portfolio de elementos.

El proceso de insertar un objeto es semejante para todos los elementos. Se debe seleccionar el objeto y arrastrarlo a la página donde se quiere implantar. Una vez ubicado, se le otorga el tamaño deseado con los nodos del elemento.

Una vez tenemos el tamaño y la ubicación del elemento, en la pestaña de propiedades se configura el objeto. Aquí se procederá a parametrizar la acción que deba realizar, así como implementar el diseño y las características visuales que deba tener.



Figura 114 - Panel Builder. Programación 2

Configuración de objetos

Los objetos gráficos, los elementos y las paginas pueden configurarse de diferentes maneras. En la pestaña de “Propierties”, podremos seleccionar los colores del diseño o las características parametrizables de cada uno. En esta pestaña también se asignarán las acciones a realizar, los eventos o las variables asociadas.

La pestaña de “Propierties” se ordena en filas agrupadas por temática. Cada una de ellas dispone de un desplegable, en el cual, seleccionaremos la opción deseada. En su defecto, cuando son parámetros abiertos, tendremos un campo alfanumérico para indicar el valor deseado.

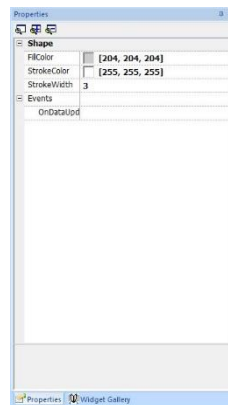


Figura 115 - Panel Builder. Programación 3

Botones, “switches” e indicadores

Uno de los grupos de elementos más utilizados son los botones e indicadores. Estos elementos se insertan del mismo modo en el programa, y todos ellos son visualmente configurables. A modo de programación funcionan del mismo modo. Se asocia a una variable y esta es modificada al presionar el botón. Los indicadores cambiarán de estado al modificarse la variable asociada. Estos elementos se asocian principalmente a una variable booleana, sin embargo, pueden programarse otras opciones.



Figura 116 - Panel Builder. Programación 4

Cuando se haya instalado el botón en la pantalla, se debe configurar. Para ello abrimos la pestaña de “Propiedades”. En ella, dispondremos el objeto del color y diseño deseado. Asimismo, si se trata de un botón será posible su configuración como interruptor o pulsado, modificando el parámetro “Click Type”

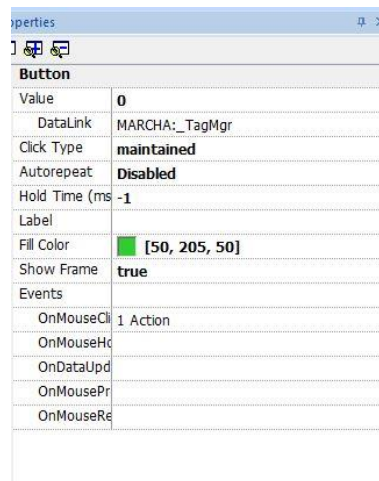


Figura 117 - Panel Builder. Programación 5

Si se pretende asociar el botón, switch o indicador a una variable, indicaremos el valor en el apartado “Value”, y a continuación, en el mismo apartado presionaremos la tecla “+” que aparece al posicionarse sobre la fila. Aquí asociaremos la variable. Cuando hayamos presionado, se nos abrirá un cuadro de dialogo donde aparecen todas las variables declaradas. Seleccionaremos la variable que se desea asociar y en la parte inferior de la ventana, indicaremos si se trata de una acción de solo lectura, escritura o ambas. Recordar que si se selecciona una acción de tipo escritura la comunicación del panel debe ser en rol de servidor o maestro.

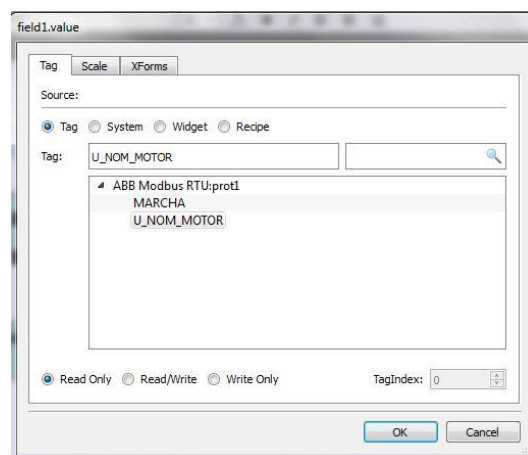


Figura 118 - Panel Builder. Programación 6

Una de las funcionalidades de los botones es la navegabilidad dentro del entorno HMI, es decir, vincular una acción a una nueva página. Para ello iremos al grupo de valores de “Events”, y seleccionaremos la opción “OnMouseClicked”, y seguidamente, “Configure Actions”.

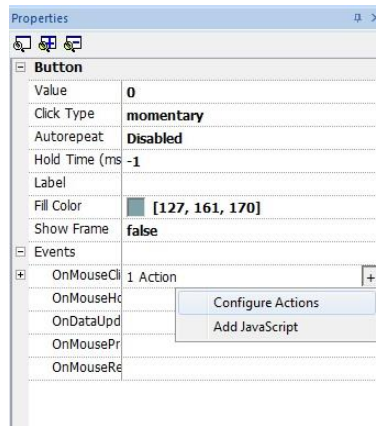


Figura 119 - Panel Builder. Programación 7

En la venta de dialogo, aparecerán todas las acciones que pueden realizarse. Están ordenadas por el tipo de elemento con el que se relacionan. Buscaremos el apartado de “Pages” y seleccionaremos “Load Page”. En la parte derecha seleccionaremos la página que deseamos cargar al presionar el botón. En el caso que queramos cargar un cuadro de dialogo el proceso es semejante y deberemos dirigirnos al apartado de “Dialogs”.

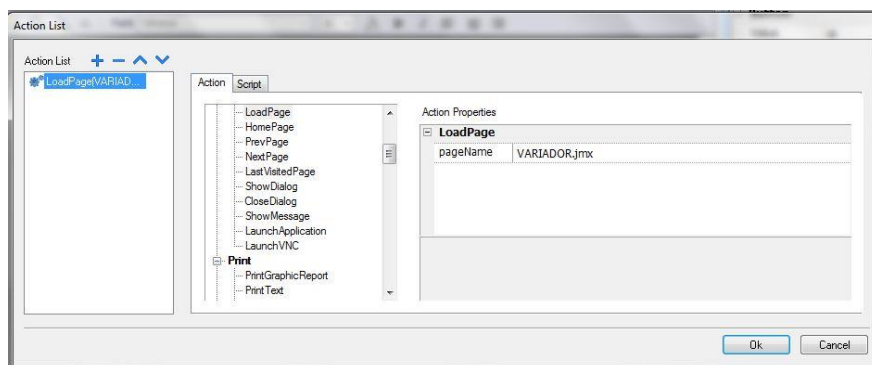


Figura 120 - Panel Builder. Programación 8

Visualización y modificación de variables

La visualización de parámetros y variables es un elemento clave en los sistemas HMI y de supervisión. Los parámetros pueden visualizarse de forma alfanumérica, o bien, mediante elementos gráficos o indicadores de reloj entre otros. Es importante, que las variables críticas del sistema se visualicen rápidamente y de forma clara, por tanto, optar por elementos gráficos puede mejorar la satisfacción y eficiencia del sistema. Nos dirigiremos a la biblioteca de objetos para instalar el elemento deseada.

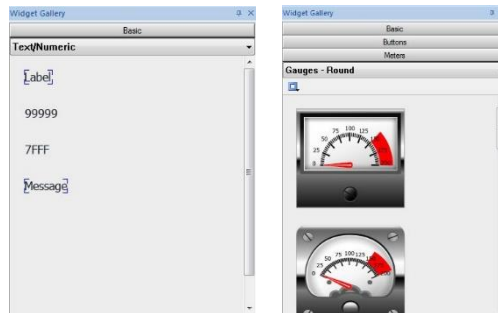


Figura 121 - Panel Builder. Programación 9

Cuando se ha insertado el objeto, sea gráfico o alfanumérico, procedemos a su configuración. Asignaremos la variable del mismo modo que se ha realizado anteriormente con los botones y si es necesario le otorgaremos un valor. Es necesario establecer los límites de la variable, ya que una modificación de un parámetro incorrectamente puede poner en peligro el sistema. Del mismo modo, si se trata de un elemento de lectura, unos límites muy amplios pueden producir una lectura imprecisa o al contrario que la lectura se encuentre fuera de límites.

Al asignar la variable, seleccionaremos lectura si solo queremos visualizar el valor de la variable o escritura si se pretende modificarla desde este objeto. En caso de modificar variables con un objeto alfanumérico, se deberá seleccionar el teclado que se ajuste a las necesidades de dicha modificación, o bien, crear un teclado nuevo.

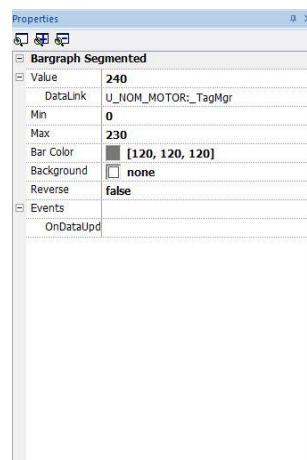


Figura 122 - Panel Builder. Programación 10

Gráficos e histogramas

La visualización de gráficas es clave en las herramientas de supervisión. Nos permite conocer el estado del sistema y detectar posibles funcionamientos anómalos, ineficientes o incidencias.

Para insertar una gráfica nos dirigimos a la galería de objetos. Se dispone de distintos modelos. En el presente proyecto se visualiza la velocidad y se ha optado por una gráfica en tiempo real, pero existen otras como históricas o diagramas de diferente tipo.

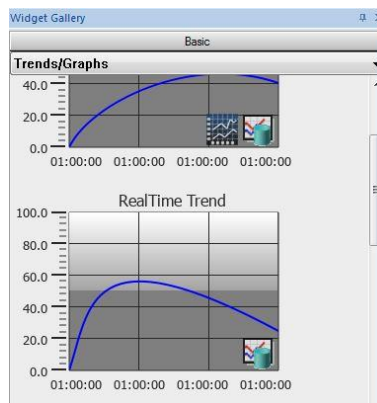


Figura 123 - Panel Builder. Programación 11

La configuración de este objeto permite gran variedad de diseños visuales y la posibilidad de visualizar distintas curvas en una misma gráfica. Pero los elementos de configuración básicos son la asignación de las variables a cada curva y la definición de los ejes.

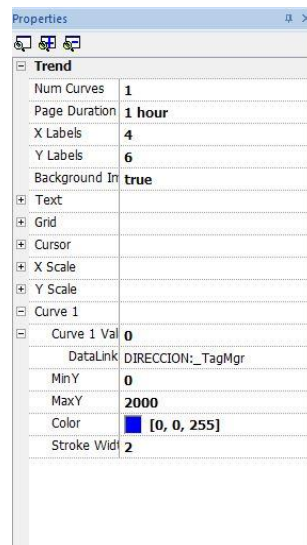


Figura 124 - Panel Builder. Programación 12

Fecha y hora

El software Panel Builder nos permite situar en la pantalla HMI un elemento de fecha y hora. Puede ser escrito o en forma de reloj. Insertaremos el elemento de la misma manera que los anteriores.

Los datos de fecha y hora pueden asociarse al reloj interno del panel, o bien a una variable de tiempo, la cual, puede indicar por ejemplo, el tiempo de un proceso o la fecha de inicio del sistema.

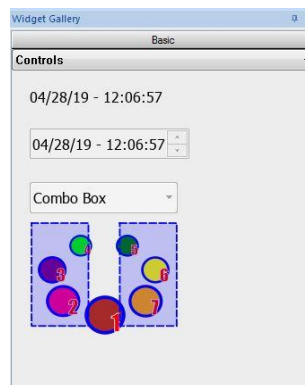
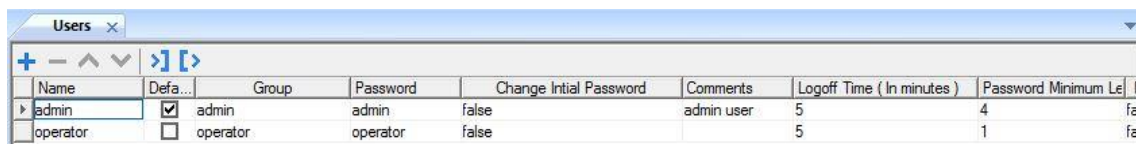


Figura 125 - Panel Builder. Programación 13

Gestión de usuarios

La seguridad de un sistema HMI es vital para el correcto funcionamiento de una instalación. Un panel de control que sea capaz de modificar el flujo del sistema, debe estar protegido y proporcionar acceso solo a las personas autorizadas.

El software Panel Builder permite la gestión de usuarios. En el apartado “Security” del árbol de proyecto, se pueden crear diferentes usuarios con niveles distintos de autorización. Estos niveles de autorización son editables.



Name	Default	Group	Password	Change Initial Password	Comments	Logoff Time (In minutes)	Password Minimum Length	Password Maximum Length
admin	<input checked="" type="checkbox"/>	admin	admin	false	admin user	5	4	fa
operator	<input type="checkbox"/>	operator	operator	false		5	1	fa

Figura 126 - Panel Builder. Programación 14

En el presente proyecto se ha creado un usuario administrador que da acceso a todo el sistema. Además, se ha configurado una desconexión automática a los cinco minutos de inacción.

Para iniciar sesión será necesario autenticarse, y se proporciona un botón asociado a la acción de desconexión. Para ello en la pestaña “Properties”, en el evento “OnClick” se ha seleccionado la opción “LogOut”. Con este sistema el panel HMI no permitirá ninguna modificación sin haberse autenticado anteriormente.

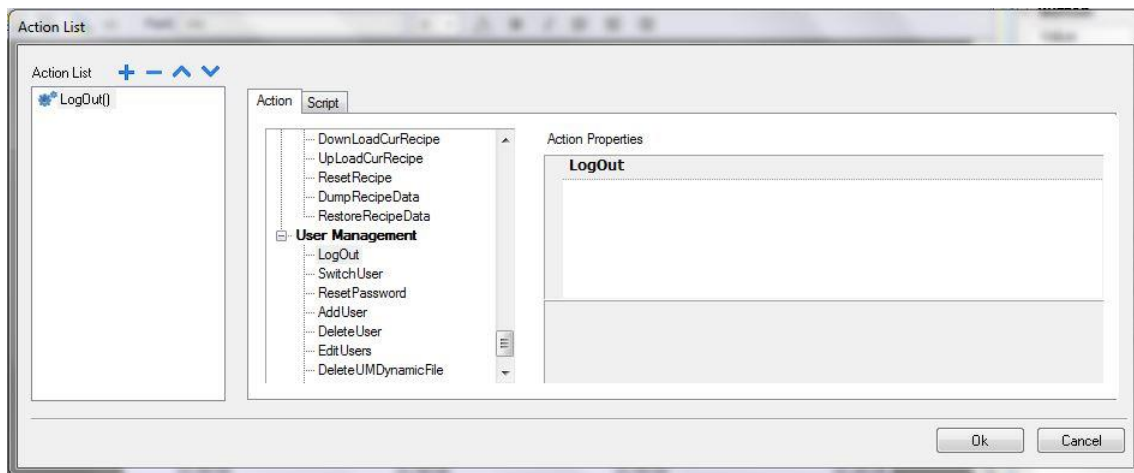


Figura 127 - Panel Builder. Programación 15

Multilenguaje

Panel Builder dispone una herramienta que facilita la traducción y simultaneidad de idiomas para el diseño del Panel HMI. La herramienta consiste en declarar los distintos idiomas que deseamos introducir en nuestro panel, y a continuación, el software indica una lista de todas las palabras empleadas. En esta lista tendremos las columnas con cada idioma, donde deberemos introducir la traducción.

Para hacer uso de esta funcionalidad, debemos dirigirnos a la opción “MultiLanguage” en el árbol de proyecto.

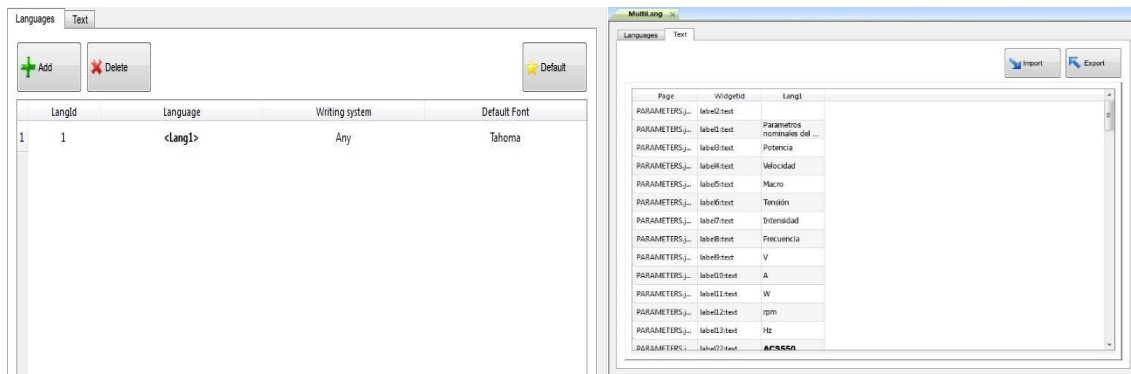



Figura 128 - Panel Builder. Programación 16

Con este sistema, se puede insertar un botón con el evento de cambio idioma, con lo cual, el propio software sustituiría las palabras indicadas en la lista anterior. Este proceso permite realizar un mismo software para diferentes idiomas, sin la necesidad de crear páginas para cada idioma, lo cual, simplifica el trabajo, a la vez que reduce el tamaño del programa, y por tanto, se aumenta en velocidad de procesamiento.

Modo simulación

El software Panel Builder, del mismo modo que otros entornos de programación, permite la opción de simular el programa para comprobar su correcto funcionamiento y localizar posibles errores.

Para acceder al modo simulación debemos presionar el botón  de la barra de tareas. El software guardará automáticamente el trabajo y abrirá una ventana donde ejecutará el programa. El modo simulación se abrirá con el tamaño indicado del panel. En entornos de programación visual es importante comprobar que los objetos y los elementos gráficos se ajustan correctamente al tamaño del dispositivo.

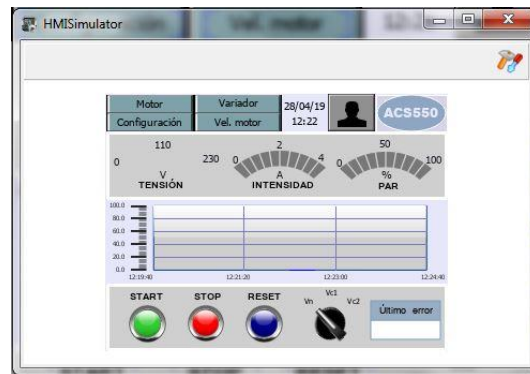


Figura 129 - Panel Builder. Programación 17

7.4.5. Descargar el programa

Una vez finalizado el programa, se debe descargar en el panel CP630. Puede realizarse a través de Ethernet y su correspondiente IP, o bien, por puerto USB. En este caso, siendo limitado el número de puertos, se ha optado por realizar la descarga a través del puerto USB. Puesto que el puerto Ethernet está siendo empleado para la comunicación con el autómata.

Para realizar la descarga del programa por puerto USB, en primer lugar, es necesario descargar el paquete con la totalidad del programa. Para ello nos dirigimos al menú “Run” y seleccionamos la opción “Manage Target”.

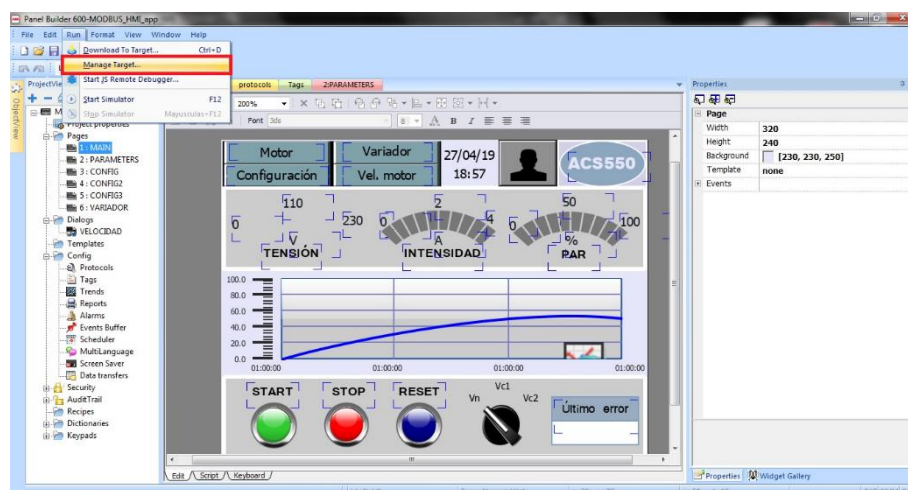


Figura 130 - Panel Builder. Descargar programa en panel CP630 1

En la ventana de dialogo seleccionamos “Update Package”, y a continuación, disponemos la ubicación donde se debe guardar el archivo. Se selecciona una memoria USB para realizar el traspaso del archivo. Es importante, cerciorarse de tener seleccionado las

opciones “Project” y “HMI Runtime & Plug-In”, para que en el paquete de instalación se disponga del programa y de los archivos de inicio.

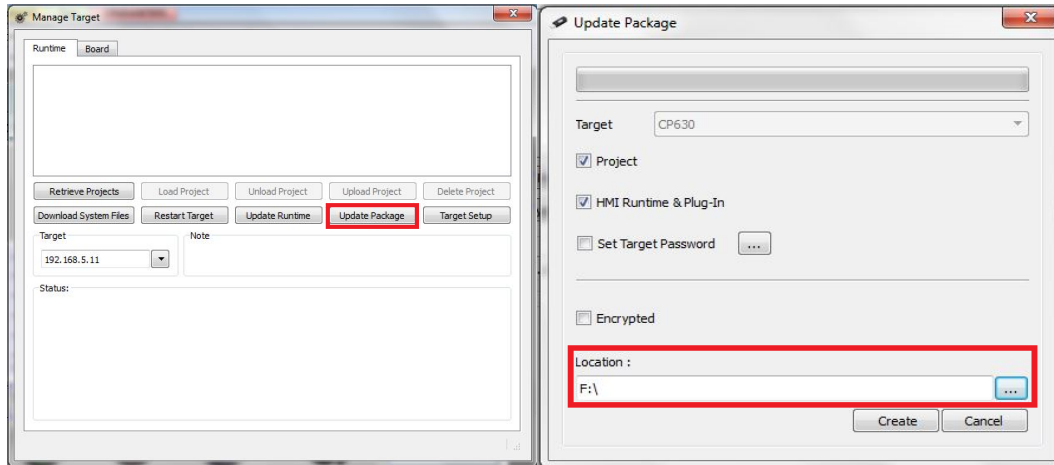


Figura 131 - Panel Builder. Descargar programa en panel CP630 2

Una vez tenemos el archivo en la memoria USB, se conecta al panel y se enciende. En la pantalla de inicio debemos seleccionar “Enter config mode”. A continuación, se introduce el usuario y contraseña de acceso.

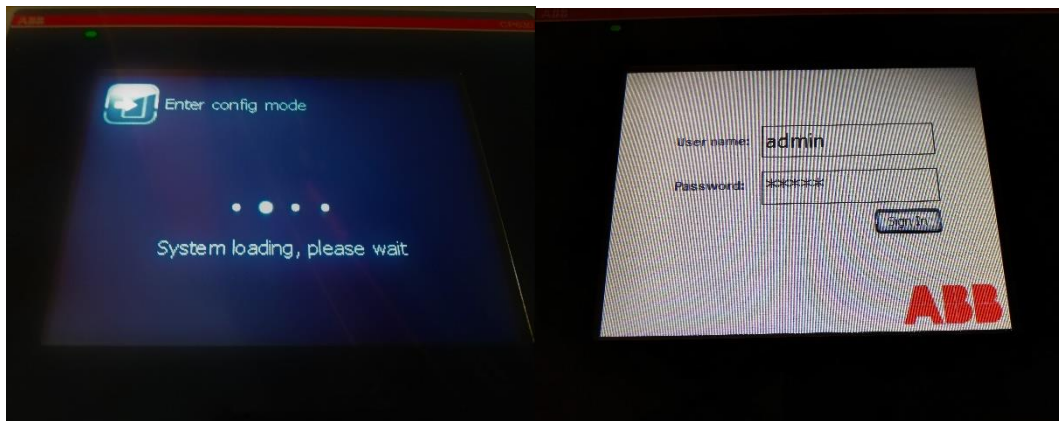


Figura 132 - Panel Builder. Descargar programa en panel CP630 3

Una vez indicadas las claves de acceso, entraremos en la pantalla de configuración. Se debe pulsar de manera prolongada en cualquier parte de la pantalla, para acceder a las opciones del menú, desde donde seleccionaremos la opción “Update”

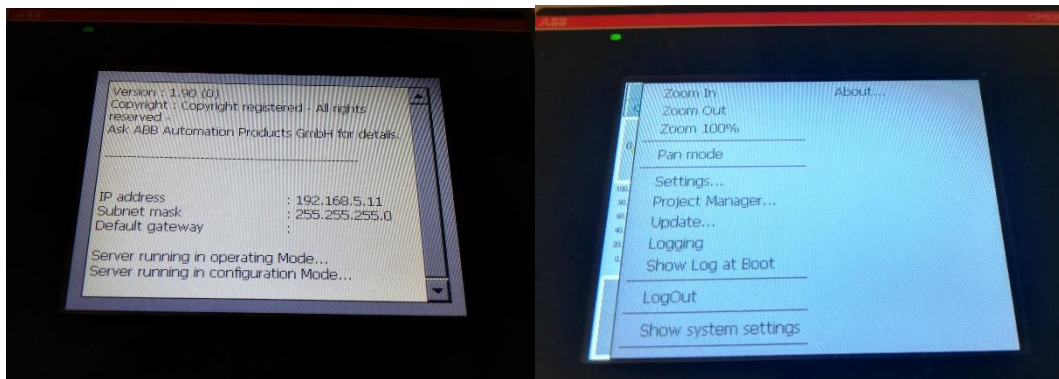


Figura 133 - Panel Builder. Descargar programa en panel CP630 4

En la nueva ventana, clicaremos en la opción “Auto select best match” y presionaremos en “Next”. Este proceso, solo se puede realizar si en la memoria USB, solo se dispone de un paquete de instalación, ya que el programa lo detectará automáticamente. En caso, de disponer de varios paquetes de instalación, se deberán seleccionar los archivos de instalación manualmente.

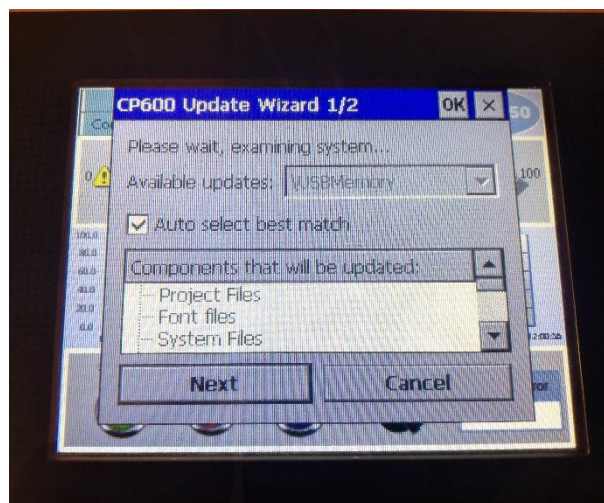


Figura 134 - Panel Builder. Descargar programa en panel CP630 5

Finalmente, el panel realizará el proceso de instalación. Una vez finalizado, el panel se reiniciará cargando automáticamente el software instalado.

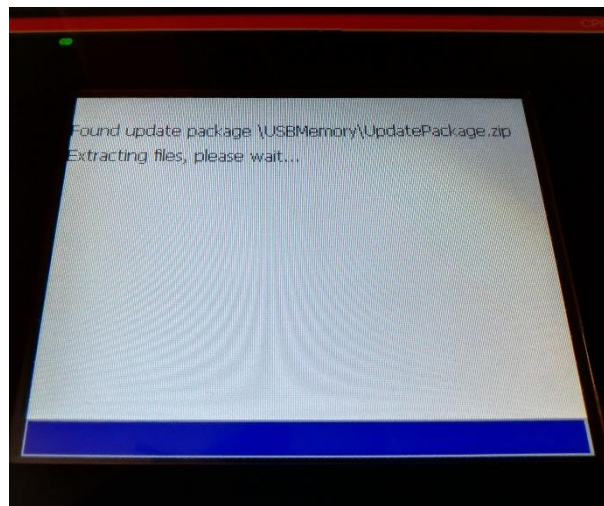


Figura 135 - Panel Builder. Descargar programa en panel CP630 6

8. Comunicació

La comunicació del sistema se realitza a través del protocol de comunicació Modbus RTU. En este apartado se sientan las bases teóricas del protocolo y la configuración de los distintos dispositivos del sistema.

La configuración del bus de campo se realiza entre los dispositivos de control del sistema:

- Convertidor ACS550
- PAC PM573-ETH
- Panel HMI CP630

Se asignan los roles de comunicació para cada uno de los dispositivos y se ajustan los parámetros de comunicació del bus de campo.

8.1.Modbus RTU

Modbus es el protocolo de comunicaciones más común utilizado en entornos industriales, sistemas de telecontrol y monitorización. La transmisión de información se realiza entre distintos equipos electrónicos conectados a un mismo bus.

Es un protocolo de comunicaciones situado en el nivel 7 del Modelo OSI, basado en la arquitectura maestro/esclavo (RTU) o cliente/servidor (TCP/IP), lo cual, permite usar Modbus sobre redes como Ethernet o WiFi.

Las características básicas para ser el protocolo más utilizado son:

- Es público y, por tanto, no deben pagarse royalties. Este hecho facilita el acceso a la información y a la estructura del protocolo.
- Su implementación es fácil y requiere poco desarrollo
- Maneja bloques de datos sin suponer restricciones, lo que implica cierta flexibilidad a la hora del intercambio de información. Por ejemplo, si se transmite un dato de 16bits de información su representación no está sujeta a ninguna restricción, por lo que puede tratarse de un dato tipo Word con signo o un entero con signo.

Los intercambios de mensajes pueden ser de dos tipos:

- Intercambio punto a punto, es decir, maestro esclavo (Peer to Peer)
- Mensajes difundidos, es decir, una comunicación unidireccional del maestro a todos los esclavos (Broadcast)

Por lo general, el maestro es una interfaz humano-máquina (HMI) o sistema SCADA y el esclavo es un sensor, controlador lógico programable (PLC) o controlador de automatización programable (PAC).

Cada dispositivo de la red Modbus posee una dirección única que se incluye en la trama de comunicación. Esta dirección tiene un tamaño de un byte. Esta característica del protocolo supone que habrá un único "Master" y hasta un máximo de 247 dispositivos "Slaves".

En este sentido la dirección 0 está reservada para mensajes de difusión sin respuesta.

Existen dos tipos de Modbus, el RTU con comunicación en datos binarios, y el ASCII con la comunicación por medio de caracteres ASCII, lo cual, hace la transmisión de datos más segura en los cambios de estado.

El protocolo Modbus organiza los datos en tramas. Dado que Modbus es un protocolo de nivel de aplicación, se requiere utilizarlo sobre una pila de protocolos que resuelva los temas específicos del tipo de red empleada.

A continuación, se puede observar cómo se organizan las tramas en el protocolo Modbus:

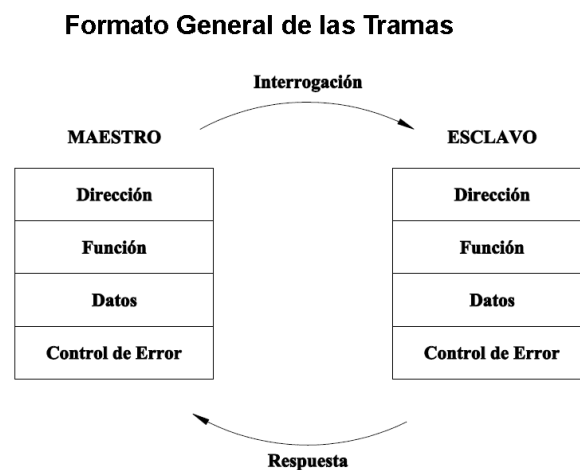


Figura 136 – Modbus RTU. Formato General de tramas [23]

Si nos fijamos en la trama existe un control del error. Esto es posible porque cada uno de los mensajes incluye información redundante que asegura su integridad en la recepción, y en consecuencia, permite la detección de errores.

Si nos centramos en el medio físico, el protocolo Modbus estaba orientado a una conectividad a través de líneas serie como pueden ser RS-232 o RS-485, pero con el paso del tiempo han aparecido variantes como la Modbus TCP, ya que Modbus se ubica en la capa de aplicación.

La conexión puede realizarse en half dúplex (RS-485 o fibra óptica) o full dúplex (RS-232). En consecuencia, el tipo de cableado podrá ser de fibra óptica, Ethernet, o bien, RJ-45 según el tipo de configuración.

El protocolo MODBUS alcanza velocidades comprendidas entre 75 y 19.200 baudios, en modo asíncrono, con una distancia máxima entre estaciones de 1200 metros sin repetidores.

8.2. Configuración convertidor

El convertidor debe configurarse para realizar la comunicación mediante Modbus RTU en su rol de esclavo o cliente. En consecuencia, deben modificarse los parámetros correspondientes a la comunicación de bus de campo.

En el caso de los convertidores ACS550, el protocolo de bus de campo encajado estándar es Modbus. Por tanto, no se deberá indicar el tipo de comunicación, puesto que ya viene configurada por defecto. En cambio, si deberán indicarse las propiedades de la comunicación que realizaremos con el autómatas. Estos parámetros deben coincidir con los configurados en Automation Builder para el COM1 (véase apartado 8.3).

Los parámetros configurados para la comunicación del bus de campo encajado son los siguientes:

Parámetro	Descripción	Valor	Descripción
9802	Sel Prot COM.	1	Comunicación Modbus mediante el canal RS485 (Configurado por defecto)
5302	ID Estación BCI	1	Estación del convertidor (debe coincidir con el valor "SLAVE" del bloque funcional ACS_COM_MOD_RTU)
5303	Vel. Trans. BCI	19,2 kb/s	Velocidad de comunicación
5304	Paridad BCI	8N1	8 bits de datos, sin paridad, 1 bit de paro
5305	Perfil control BCI	ABB DRV LIM	Funcionamiento del código de control/estado se ajusta al perfil de convertidores ABB

Tabla 13 – Parámetros básicos ACS550 para la comunicación Modbus RTU

La modificación de parámetros del bus de campo encajado, únicamente, tendrá efecto si se apaga y vuelve a encender el convertidor. (Para la modificación de parámetros del convertidor véase apartado 6.1.4). Del mismo modo, la comunicación del convertidor mediante el bus de campo encajado solo tendrá efecto si se configura el "switch" del puerto serie. (Véase apartado 6.4.2)

Conjuntamente con la configuración de los parámetros de comunicación, se deben autorizar las acciones y el acceso a los parámetros del convertidor a través del bus de campo encajado.

Los parámetros adicionales, que se han modificado, para el correcto desarrollo de la aplicación de control son los siguientes:

Parámetro	Descripción	Valor	Descripción
1001	Comandos EXT1	10	Asigna el código de comando del bus de campo como fuente para el marcha/paro y dirección.
1604	Sel Rest Fallo	8	Asigna el código de comando del bus de campo como fuente para la restauración de fallos.
2013	Sel par min	7	Asigna el código de comando del bus de campo como fuente para la selección del par mínimo.

2014	Sel par max	7	Asigna el código de comando del bus de campo como fuente para la selección del par máximo.
2201	Sel ACE/DEC	7	Asigna el código de comando del bus de campo como fuente para la selección de rampas de aceleración y deceleración.
3018	Func Fallo Com	1	Muestra un fallo de comunicación y el variador se para por si solo.

Tabla 14 – Parámetros adicionales ACS550 para la comunicación Modbus RTU

La modificación a través del bus de campo encajado de los parámetros del convertidor referentes a la configuración del modo de funcionamiento, solo pueden ser modificados con el convertidor parado. En caso contrario, los cambios no se realizarán.

8.3. Configuración PAC

El autómatas programable debe comunicarse, mediante Modbus RTU, con dos dispositivos. El rol del autómatas es diferente para cada uno de ellos. El convertidor realizará las funciones de esclavo, y por tanto, el autómatas de maestro. En cambio, el panel de control realiza funciones de maestro para el autómatas, que en esta comunicación toma el papel de esclavo.

Los parámetros de comunicación deben ser iguales para cada comunicación Modbus RTU. En el caso del presente proyecto, los valores de la parametrización Modbus son idénticos en las dos comunicaciones: PAC-Variador y Panel-PAC.

Las comunicaciones se realizan a través de puerto serie, con el estándar de comunicaciones de la capa física, RS485 (Para más información sobre el conexionado véase apartado 6.4). En este apartado veremos la configuración de parámetros de los distintos dispositivos para la comunicación Modbus RTU del autómatas programable.

Configuración variador de frecuencia

En primer lugar, debemos indicar la naturaleza de la comunicación que se desea establecer y en el puerto que se va a llevar a cabo. La comunicación del autómatas y el convertidor se realiza en la salida del autómatas "COM1".

Nos dirigimos al árbol de dispositivos de Automation Builder y hacemos clic con el botón izquierdo sobre la interface COM1. Seleccionamos "Agregar Objeto". En la ventana de dialogo, elegimos el protocolo de comunicación, en este caso Modbus. Si se tienen diversas comunicaciones en un mismo proyecto, existe la posibilidad de otorgarle un nombre para diferenciarlas. Finalmente, seleccionamos "Substituir Objeto".

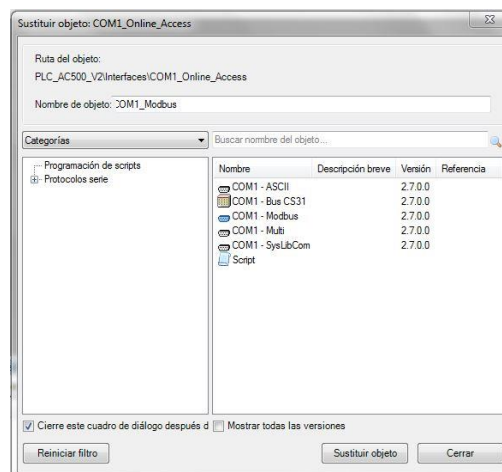


Figura 137 - Modbus RTU. Configuración Automation Builder convertidor 1

A continuación, hacemos doble clic en el mismo icono anterior "COM1", que ahora nos aparecerá con la indicación de Modbus. Se nos abre la pestaña de configuración del protocolo con los valores por defecto. Aquí se deben configurar los mismos parámetros que se han configurado en el convertidor. En este caso, solo deberemos modificar los parámetros siguientes:

- Modo RTS: Telegrama
- Paridad: Ninguno
- Modo de operación: Cliente (Este es el rol del variador y no del autómatas)

En la figura 138, es posible observar la configuración final de todos los parámetros necesarios:

COM1 - Modbus Configuración	Parámetro	Tipo	Valor	Valor predeterminado	Unidad	Descripción
Configuración de servidor Modbus	Habilitar inicio de sesión	Enumeration of BYTE	Deshabilitado	Deshabilitado		Comprobar el inicio de sesión Co
	Mando RTS	Enumeration of BYTE	Telegrama	Ninguno		Mando RTS i deberá ajustarse e
	Valor de fin del telegrama	WORD(0..65535)	3	3		Ajustar el valor de fin del telegra
	Velocidad en baudios	Enumeration of DWORD	19200	19200	Bits/s	Ajustar la velocidad en baudios e
	Paridad	Enumeration of BYTE	Ninguno	Par		Ajustar el tipo de Bit de paridad
	Bits de datos	Enumeration of BYTE	8	8	Bits/carácter	Ajustar el tamaño de carácter
	Bits de paro	Enumeration of BYTE	1	1		Ajustar la cantidad de Bits de pa
	Ejecutar con error de configuración	Enumeration of BYTE	No	No		Iniciar el programa PLC también
	Modo de operación	Enumeration of BYTE	Cliente	Ninguno		Ajustar el modo de operación
	Dirección	BYTE(0..255)	0	0		Ajustar la dirección del dispositi

Figura 138 - Modbus RTU. Configuración Automation Builder convertidor 2

Configuración panel HMI

La configuración del panel HMI se realiza a partir del mismo proceso anterior. Se establecerá la naturaleza del protocolo de comunicación y el puerto por el cual se realiza. En este caso será el “COM2”.

Del mismo modo anterior, nos dirigimos al árbol de dispositivos y hacemos clic con el botón izquierdo sobre la interface COM2. Seleccionamos “Agregar Objeto”. En la ventana de dialogo, elegimos el protocolo Modbus.

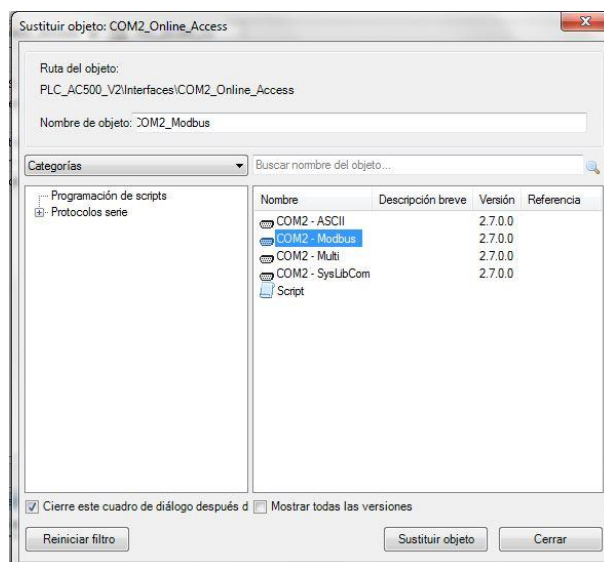
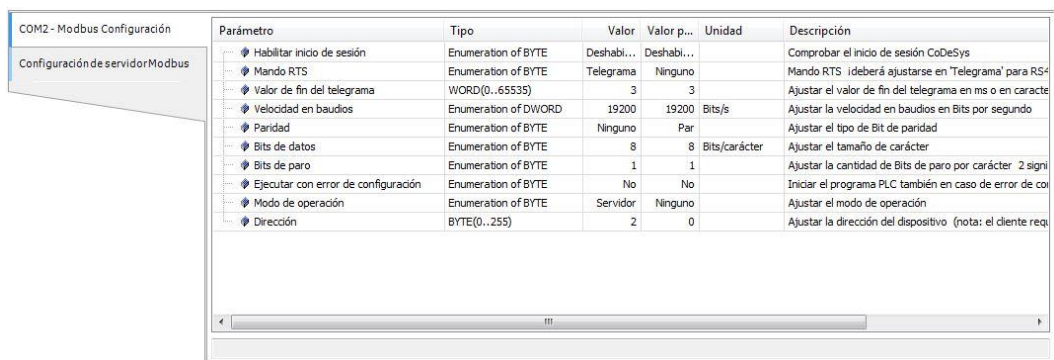


Figura 139 - Modbus RTU. Configuración Automation Builder panel HMI 1

En la pestaña de configuración del protocolo, se configuran los parámetros del bus de campo, que deberán coincidir con los parámetros configurados en el panel HMI. En este caso, solo deberemos modificar los parámetros siguientes:

- Modo RTS: Telegrama
- Paridad: Ninguno
- Modo de operación: Servidor (Este es el rol del variador y no del autómatas)
- Dirección: 2 (Debe ser diferente para cada equipo)

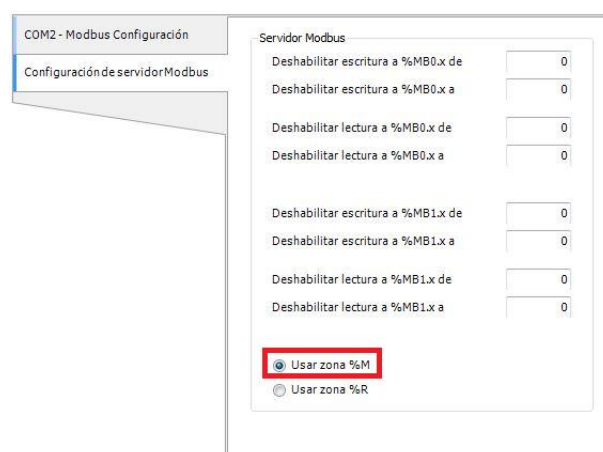
En la figura siguiente, es posible observar la configuración final de todos los parámetros necesarios:



Parámetro	Tipo	Valor	Valor p...	Unidad	Descripción
Habilitar inicio de sesión	Enumeration of BYTE	Deshabi...	Deshabi...		Comprobar el inicio de sesión CoDeSys
Mando RTS	Enumeration of BYTE	Telegrama	Ninguno		Mando RTS (deberá ajustarse en 'Telegrama' para RS485)
Valor de fin del telegrama	WORD(0..65535)	3	3		Ajustar el valor de fin del telegrama en ms o en caracteres
Velocidad en baudios	Enumeration of DWORD	19200	19200	Bits/s	Ajustar la velocidad en baudios en Bits por segundo
Paridad	Enumeration of BYTE	Ninguno	Par		Ajustar el tipo de Bit de paridad
Bits de datos	Enumeration of BYTE	8	8	Bits/carácter	Ajustar el tamaño de carácter
Bits de paro	Enumeration of BYTE	1	1		Ajustar la cantidad de Bits de paro por carácter (2 significa 2 espacios)
Ejecutar con error de configuración	Enumeration of BYTE	No	No		Iniciar el programa PLC también en caso de error de configuración
Modo de operación	Enumeration of BYTE	Servidor	Ninguno		Ajustar el modo de operación
Dirección	BYTE(0..255)	2	0		Ajustar la dirección del dispositivo (nota: el cliente requiere dirección 0)

Figura 140 - Modbus RTU. Configuración Automation Builder panel HMI 2

Finalmente, es necesario asegurarse que el protocolo de comunicación está trabajando con el tipo de memoria interna y no con los datos retentivos. Para ello, accedemos al menú "Configuración servidor Modbus" y seleccionamos la casilla "Usar zona %M".



COM2 - Modbus Configuración

Configuración de servidor Modbus

Servidor Modbus

Deshabilitar escritura a %MB0.x de: 0

Deshabilitar escritura a %MB0.x a: 0

Deshabilitar lectura a %MB0.x de: 0

Deshabilitar lectura a %MB0.x a: 0

Deshabilitar escritura a %MB1.x de: 0

Deshabilitar escritura a %MB1.x a: 0

Deshabilitar lectura a %MB1.x de: 0


Deshabilitar lectura a %MB1.x a: 0

☒ Usar zona %M

☐ Usar zona %R

Figura 141 - Modbus RTU. Configuración Automation Builder panel HMI 3

8.4. Configuración HMI

En el software Panel Builder, la configuración del protocolo de comunicación del panel se realiza en la opción “Protocols” del árbol de proyecto. En esta pestaña parametrizaremos la naturaleza del protocolo de comunicación, el puerto por el cual se realiza, y por defecto, el autómatas con el que se comunica. Abrimos la pestaña, y seleccionamos el icono .

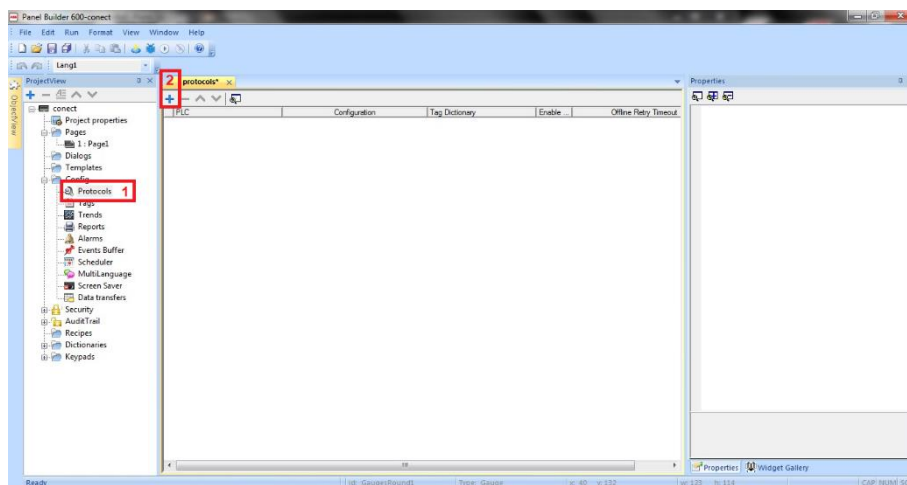


Figura 142 - Modbus RTU. Configuración Panel Builder panel HMI 1

En la nueva fila que nos aparece, en el apartado PLC seleccionaremos “ABB Modbus RTU” en el desplegable. Esta acción nos abrirá una nueva ventana de dialogo. Aquí configuraremos el “Nodo ID” que debe coincidir con la dirección establecida en el COM2 de Automation Builder (véase apartado 8.3). A continuación, seleccionaremos el tipo de autómatas en la parte inferior de la ventana. Para el autómatas del presente proyecto se selecciona la fila “ABB PM59x, PM58x, PM573”. Finalmente, clicamos en el botón de “COMM” para configurar los parámetros del protocolo.

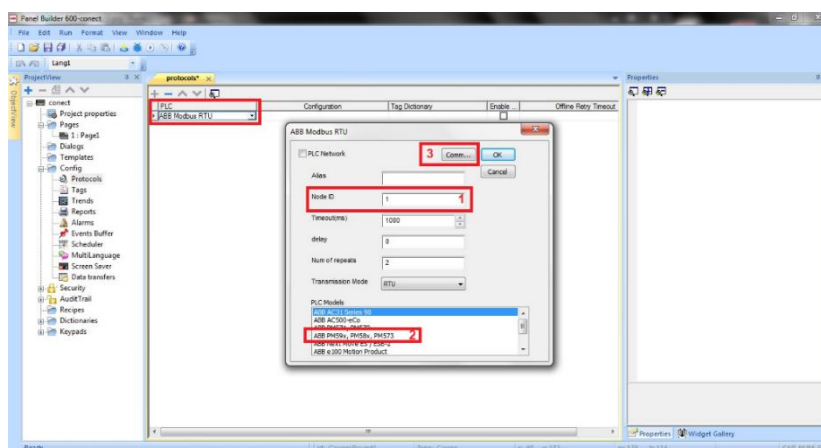


Figura 143 - Modbus RTU. Configuración Panel Builder panel HMI 2

En esta ventana deberemos introducir los valores de configuración del bus de campo, que con anterioridad han sido introducidos en Automation Builder. Los parámetros finales se pueden observar en la figura 145.

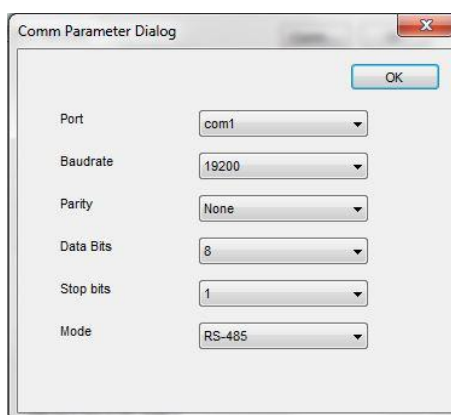


Figura 144 - Modbus RTU. Configuración Panel Builder panel HMI 3

9. Control y supervisión de un convertidor mediante PAC y HMI

El sistema que se presenta tiene la misión de controlar y supervisar un convertidor, a través de un autómatas programable y un sistema HMI. El programa tiene la capacidad de leer y modificar los parámetros más significativos del convertidor, como por ejemplo, los valores del PID para realizar un control de velocidad. Dispone, además, de un sistema de velocidades constantes para actuar en el sistema a velocidades controladas, al mismo tiempo, que permite el control de la rampa de aceleración y de par.

La comunicación entre los distintos dispositivos se realiza a partir del protocolo estándar Modbus RTU. A continuación, se presenta el código del autómatas y las pantallas del sistema HMI.

9.1. Programa y visualización PAC

Se declaran dos tipos de variables globales en el programa. Las variables necesarias para la compilación del autómatas y las variables que serán compartidas con el sistema HMI.

Variables globales del autómatas

```

0001 VAR_GLOBAL
0002
0003 (*Comunicación Modbus y control convertidor *)
0004
0005 COMUNICACION: ACS_COM_MOD_RTU;
0006 CONTROL: ACS_DRIVES_CTRL_STANDARD;
0007 ON: BOOL;
0008
0009 ModbusToken_COM1: ACS_MOD_TOKEN_TYPE;
0010 DRIVE: ACS_DRIVE_DATA_TYPE;
0011
0012 (*Lectura de parametros convertidor*)
0013
0014 P1: ACS_MOD_READ_N_PRM;
0015 P2: ACS_MOD_READ_N_PRM;
0016 P3: ACS_MOD_READ_N_PRM;
0017 P4: ACS_MOD_READ_N_PRM;
0018 P5: ACS_MOD_READ_N_PRM;
0019 P6: ACS_MOD_READ_N_PRM;
0020 P7: ACS_MOD_READ_N_PRM;
0021 P8: ACS_MOD_READ_N_PRM;
0022 P9: ACS_MOD_READ_N_PRM;
0023 P10: ACS_MOD_READ_N_PRM;
0024 P11: ACS_MOD_READ_N_PRM;
0025 P12: ACS_MOD_READ_N_PRM;
0026

```

Figura 145 – Automation Builder: Programa PAC control y supervisión 1

```

0027 (*Modificación de parametros convertidor*)
0028
0029 MP_1: ACS_MOD_WRITE_N_PRM;
0030 MP_2: ACS_MOD_WRITE_N_PRM;
0031 MP_3: ACS_MOD_WRITE_N_PRM;
0032 MP_4: ACS_MOD_WRITE_N_PRM;
0033 MP_5: ACS_MOD_WRITE_N_PRM;
0034 MP_6: ACS_MOD_WRITE_N_PRM;
0035 MP_7: ACS_MOD_WRITE_N_PRM;
0036 MP_8: ACS_MOD_WRITE_N_PRM;
0037 MP_9: ACS_MOD_WRITE_N_PRM;
0038 MP_10: ACS_MOD_WRITE_N_PRM;
0039 MP_11: ACS_MOD_WRITE_N_PRM;
0040 MP_12: ACS_MOD_WRITE_N_PRM;
0041 MP_13: ACS_MOD_WRITE_N_PRM;
0042 MP_14: ACS_MOD_WRITE_N_PRM;
0043 MP_15: ACS_MOD_WRITE_N_PRM;
0044
0045 (*Lectura de parametros nominales del motor*)
0046
0047 MOTOR1: ACS_MOD_READ_N_PRM;
0048 MOTOR2: ACS_MOD_READ_N_PRM;
0049 MOTOR3: ACS_MOD_READ_N_PRM;
0050 MOTOR4: ACS_MOD_READ_N_PRM;
0051 MOTOR5: ACS_MOD_READ_N_PRM;
0052 MOTOR6: ACS_MOD_READ_N_PRM;
0053
0054
0055
0056 END_VAR

```

Figura 146 - Automation Builder: Programa PAC control y supervisión 2

Variables globales exportadas al sistema HMI

```

0001 VAR_GLOBAL
0002
0003 (*Variables de control*)
0004
0005 MARCHA AT%MX0.0.1: BOOL;
0006 PARO AT%MX0.0.2: BOOL;
0007 RESET AT%MX0.0.3: BOOL;
0008 MOD_PARAM AT%MX0.0.4: BOOL;
0009 ED1 AT%MX0.0.5: BOOL;
0010 ED2 AT%MX0.0.6: BOOL;
0011 EXECUTE AT%MX0.0.7: BOOL;
0012
0013
0014 VELOCIDAD AT%MW0.1: INT:=10000; (* -20000,0 a +20000,0*)
0015
0016
0017 (*Variables de lectura*)
0018
0019 DIRECCION AT%MW0.2: INT;
0020 BUS_CC AT%MW0.3: INT;
0021 TEMP_ACS550 AT%MW0.4: INT;
0022 KWH AT%MW0.5: INT;
0023 ANGULO AT%MW0.6: INT;
0024 POTENCIA AT%MW0.7: INT;
0025 PAR AT%MW0.8: INT;
0026 INTENSIDAD AT%MW0.9: INT;
0027 FREQ_SALIDA AT%MW0.10: INT;
0028 TENSION_SALIDA AT%MW0.10: INT;
0029 FALLO AT%MW0.12: INT;
0030 MACRO AT%MW0.13: INT;

```

Figura 147 - Automation Builder: Programa PAC control y supervisión 3

```

0031
0032
0033 U_NOM_MOTOR AT%MW0.14: INT;
0034 I_NOM_MOTOR AT%MW0.15: INT;
0035 F_NOM_MOTOR AT%MW0.16: INT;
0036 V_NOM_MOTOR AT%MW0.17: INT;
0037 P_NOM_MOTOR AT%MW0.18: INT;
0038 COSFI_NOM_MOTOR AT%MW0.19: INT;
0039
0040 (*Variables de modificación*)
0041
0042 V_MIN AT%MW0.20: INT:=0;
0043 V_MAX AT%MW0.21: INT:=18000;
0044 PAR_MIN AT%MW0.22: INT:=3000;
0045 PAR_MAX AT%MW0.23: INT:=3000;
0046 T_ACEL AT%MW0.24: INT:=50;
0047 T_DES AT%MW0.25: INT:=50;
0048 RAMPA AT%MW0.26: INT:=0; (* 0,0 a 1000,0 s *)
0049 T_ACE_PAR AT%MW0.27: INT:=0;
0050 T_DES_PAR AT%MW0.28: INT:=0;
0051 F_CONMUTACION AT%MW0.29: INT:=2;
0052 I_MAX AT%MW0.30: INT:=40;
0053 GANANCIA_PROP AT%MW0.31: INT:=500;
0054 T_INTEGRACION AT%MW0.32: INT:=50;
0055 T_DERIVACION AT%MW0.33: INT:=0;
0056 COMP_ACE AT%MW0.34: INT:=0;
0057
0058 END_VAR

```

Figura 148 - Automation Builder: Programa PAC control y supervisión 4

POU's

El programa del autómatas se sustenta en cuatro unidades de programación (POU's), en las cuales, se utilizan los lenguajes FBD, LADDER y ST. Todos ellos amparados por las directrices del estándar IEC-6113-3.



Figura 149 - Automation Builder: Programa PAC control y supervisión 5

- POU principal (PLC PGR)

Es la unidad principal del sistema, se encarga de llamar a los distintos subprogramas que realizan la carga del sistema. En este caso, se accionarán la POU del convertidor y el marcha/paro del sistema. Las velocidades constantes se accionan desde el programa del convertidor.

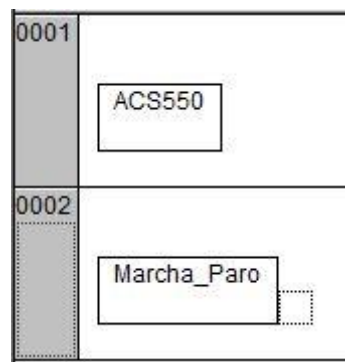


Figura 150 - Automation Builder: Programa PAC control y supervisión 6

- Marcha / Paro

El programa del marcha-paro es un pequeño subprograma encargo de activar la entrada del bloque funcional para el control del convertidor. Este sistema es necesario para poder actuar desde el panel HMI con un botón tipo pulsador. Por tanto, tener un botón independiente de paro. Este subprograma se escribe en lenguaje LADDER.



Figura 151 - Automation Builder: Programa PAC control y supervisión 7

- Velocidades constantes

El sistema tiene la capacidad de parametrizar la velocidad del motor. Además, dispone de la posibilidad de actuar en el sistema, a partir de dos velocidades constantes. Si se accionan alguna de las entradas que activan estas velocidades constantes, la velocidad parametrizada no se tendrá en cuenta y tendrá prioridad la velocidad constante. En caso, de activarse las dos entradas a la vez, el sistema entenderá esta acción como un error, y priorizará la velocidad menor.

El programa se escribe en lenguaje estructurado, y actúa como bloque funcional propio en el programa del convertidor.

```

0001 PROGRAM V_K
0002 VAR_INPUT
0003   EN:BOOL;
0004   EN2:BOOL;
0005   INPUT:INT;
0006 END_VAR
0007
0008 VAR_OUTPUT
0009   SPEEDREF:INT;
0010 END_VAR
0011
0012 VAR
0013   VEL:INT;
0014 END_VAR
0015
0001 VEL:=INPUT;
0002 IF EN=TRUE THEN
0003   OUTPUT:=5000;
0004 ELSIF EN2=TRUE THEN
0005   OUTPUT:=15000;
0006 ELSIF EN=FALSE AND EN2=FALSE THEN
0007   OUTPUT:=VEL;
0008 END_IF

```

Figura 152 - Automation Builder: Programa PAC control y supervisión 8

- POU ACS550

La unidad de programación ACS550, es la encargada de comunicarse con el convertidor y actuar sobre él. Dispone de tres partes diferenciadas. En la primera parte, se realiza la comunicación vía Modbus RTU y el control del autómat, a continuación, tenemos un grupo de lectura de parámetros y, finalmente, un grupo de modificación de parámetros

En la parte superior se disponen del módulo de comunicación Modbus RTU y el bloque de control. En este bloque de control se dispone del módulo de velocidades constantes que actúa sobre la entrada de velocidad de referencia.

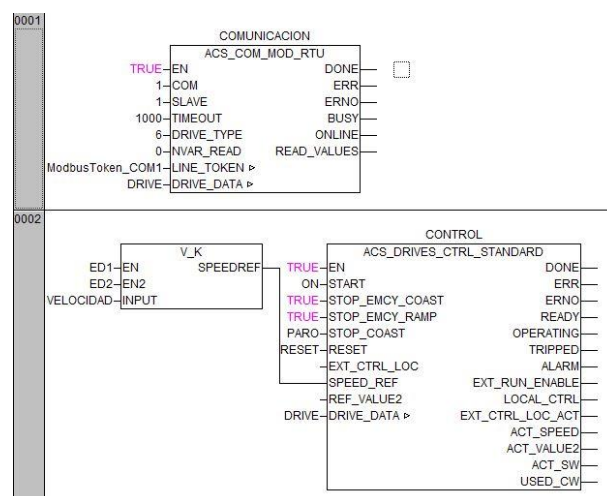


Figura 153 - Automation Builder: Programa PAC control y supervisión 9

A continuació, tenemos los módulos de lectura de los distintos parámetros.

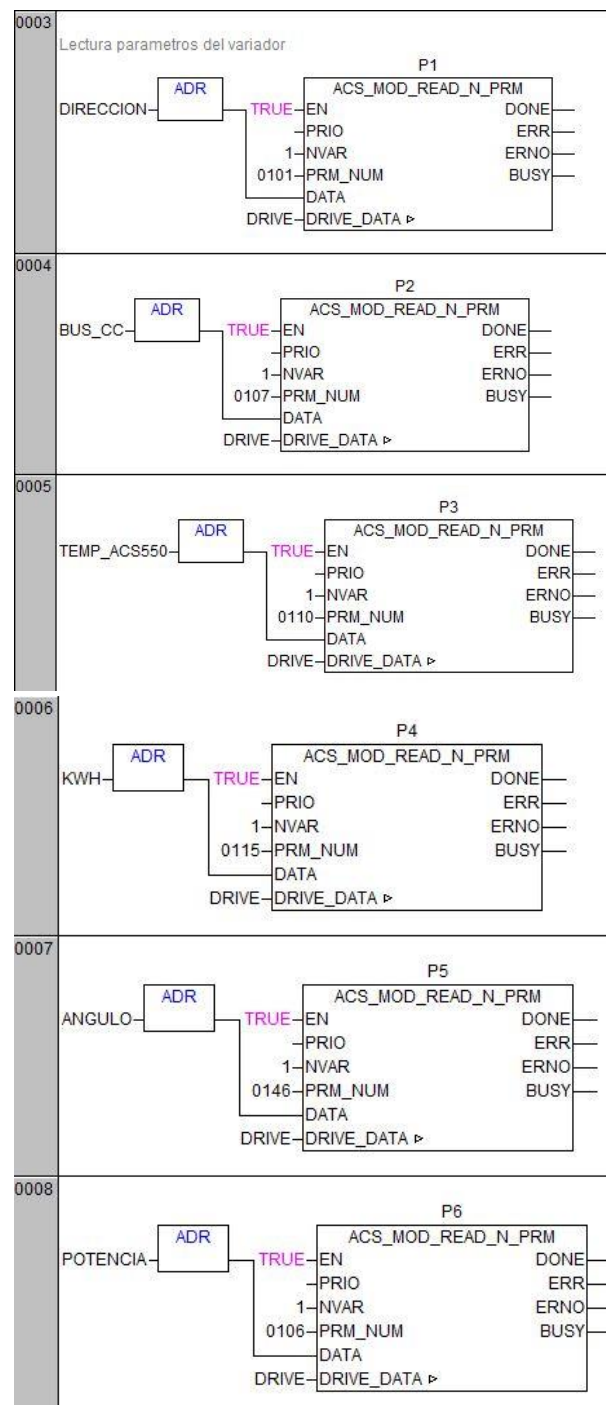


Figura 154 - Automation Builder: Programa PAC control y supervisión 10

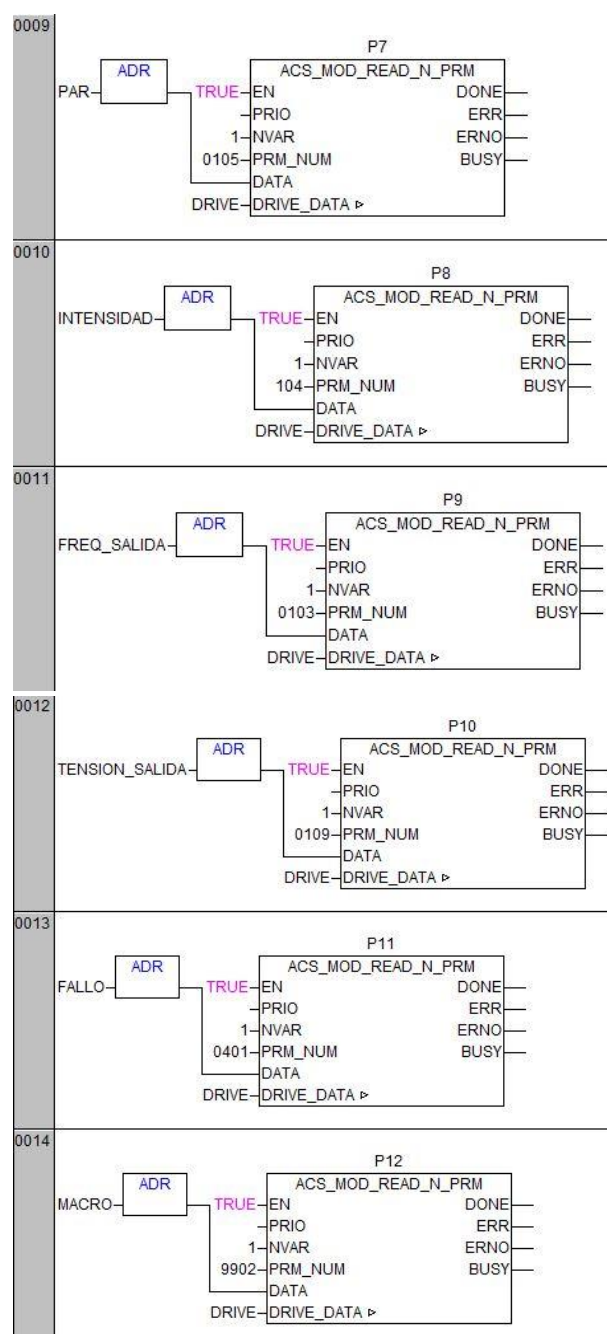


Figura 155 - Automation Builder: Programa PAC control y supervisión 11

En la parte inferior, se disponen los módulos de modificación de parámetros.

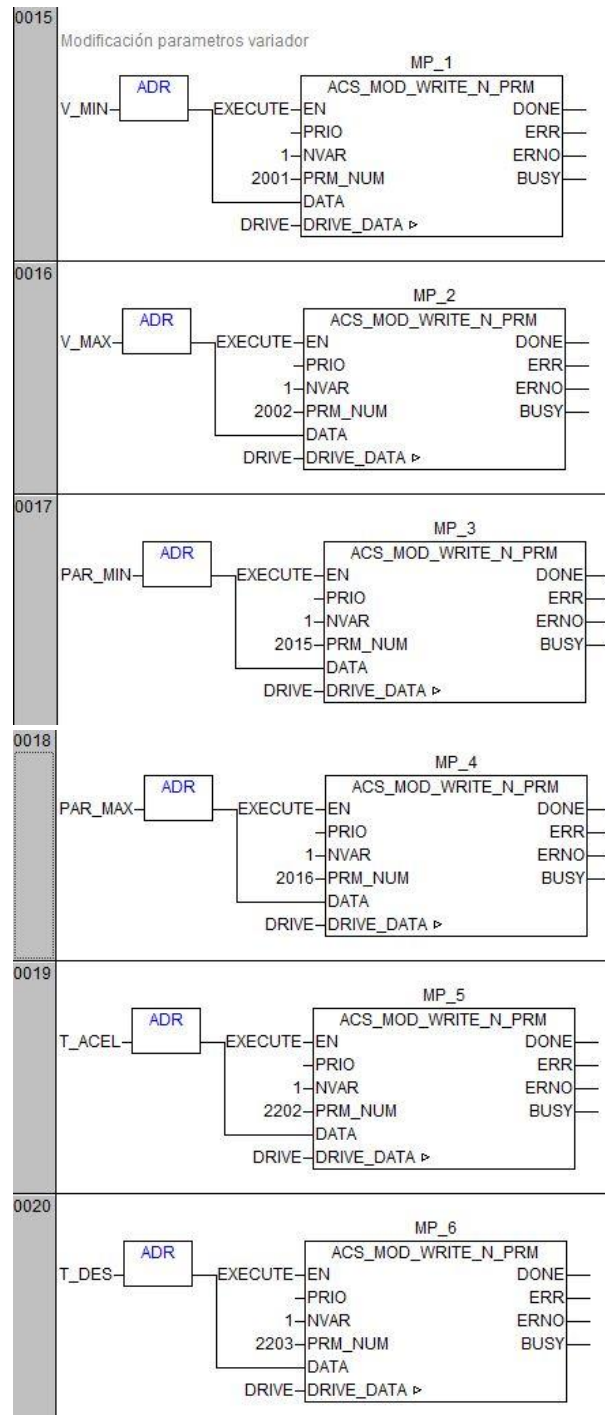


Figura 156 - Automation Builder: Programa PAC control y supervisión 12

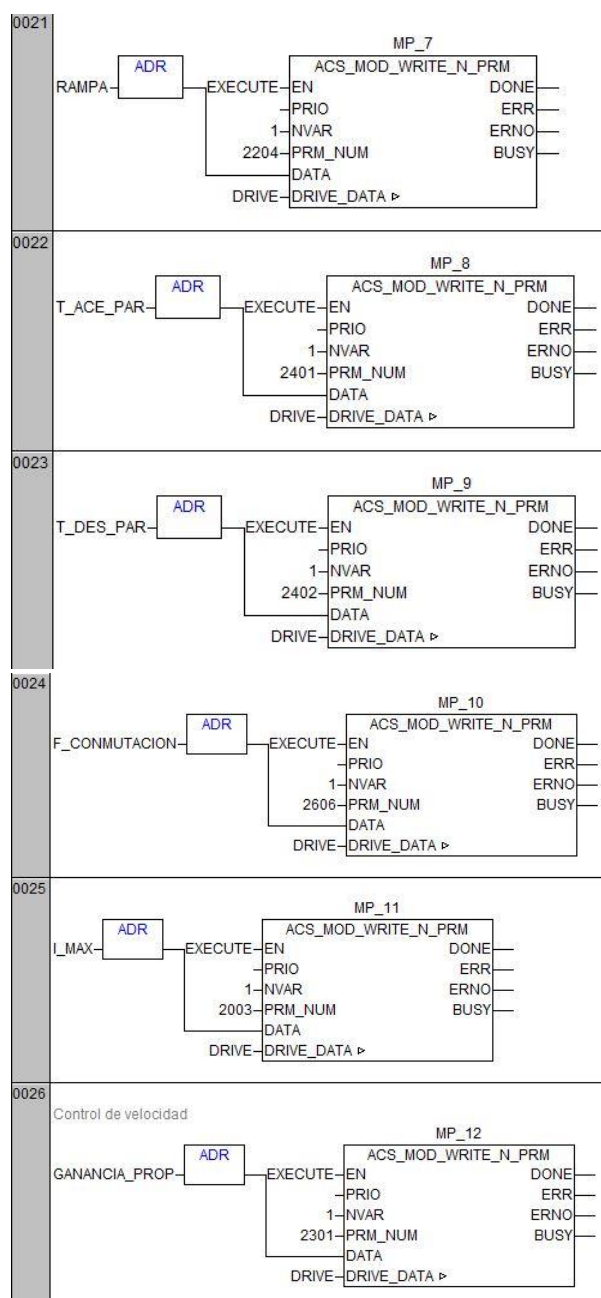


Figura 157 - Automation Builder: Programa PAC control y supervisión 13

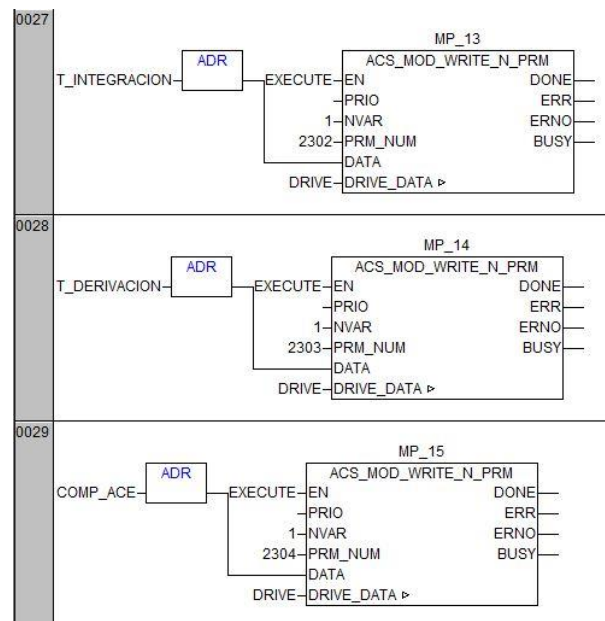


Figura 158 - Automation Builder: Programa PAC control y supervisión 14

En un apartado adicional en la parte inferior, se disponen los parámetros de lectura referentes a los valores nominales del motor. Se sitúan en la parte inferior, para diferenciarlos de forma clara, de los parámetros que afectan al convertidor.

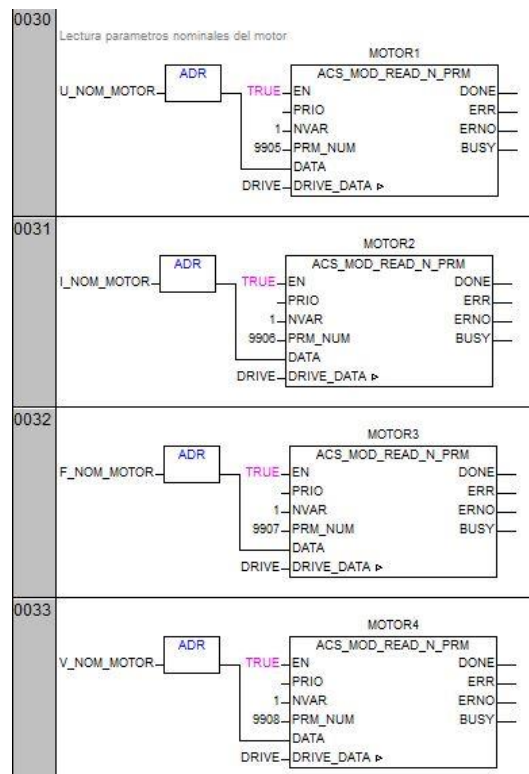


Figura 159 - Automation Builder: Programa PAC control y supervisión 15

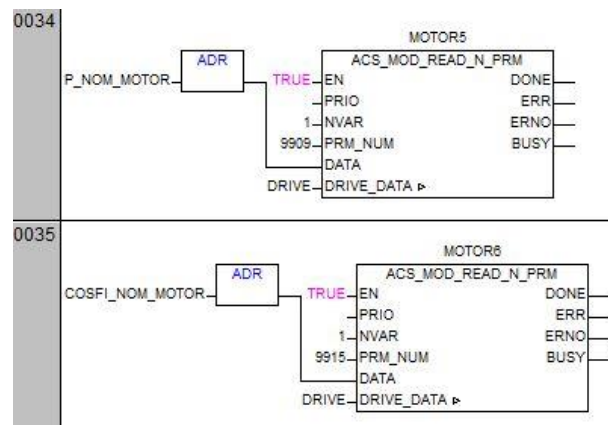


Figura 160 - Automation Builder: Programa PAC control y supervisión 16

Visualización

Se dispone de una visualización en CODESYS para el desarrollo de los tests y la depuración del sistema. En la parte superior, se visualizan los parámetros importantes del sistema, diferenciados dos apartados: los valores nominales del motor y los parámetros del convertidor.

En la parte central se sitúa una gráfica de la velocidad del motor en tiempo real.

En la parte inferior, se sitúan los parámetros modificables del convertidor.

En la parte derecha de la visualización, se encuentra un pequeño panel de control. En este cuadro de control, se puede actuar sobre las velocidades variables y accionar el motor que dispone de un indicador de marcha. Además, se puede observar el ultimo error producido en el sistema.

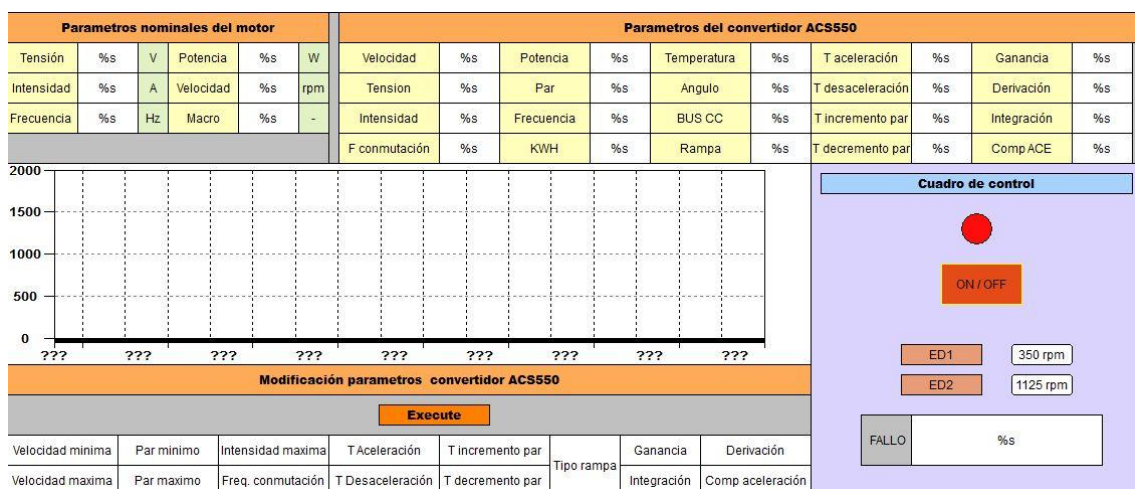


Figura 161 - Automation Builder: Programa PAC control y supervisión. Visualización

9.2.HMI: Interfaz de usuario

El sistema HMI de control y supervisión se compone de diversas pantallas. En la pantalla inicial, se dispone de un panel de control en la parte inferior. En ella se puede acceder a las velocidades constantes, accionar y resetear el sistema.

En la parte central, se encuentra una gráfica en tiempo real de la velocidad del sistema, donde además se puede observar la velocidad consigna. En la parte inmediatamente superior, se sitúan tres medidores analógicos para supervisar variables críticas del sistema.

La parte superior está reservada para la navegación dentro del programa HMI. Desde aquí se accederán a las diversas pantallas y al acceso de usuarios.



Figura 162 – Panel Builder: Programa control y supervisión HMI 1

Las variables que se muestran como elemento de supervisión y las variables de modificación han sido ajustadas para mostrar las unidades correctas, corrigiendo así, el valor entero que entrega el protocolo estándar Modbus RTU.

En el botón “Vel. Motor”, accedemos a un cuadro de dialogo, en el cual, se puede indicar la velocidad consigna del sistema. Se ha considerado realizar las modificaciones de velocidad a través de este sistema por motivos de seguridad. La velocidad es el parámetro más crítico del sistema y, por tanto, no se debe tener acceso desde una pantalla principal. De este modo, se consigue que el usuario cambie la velocidad, de manera consciente, después de realizar algún paso anterior.



Figura 163 - Panel Builder: Programa control y supervisión HMI 2

En la pantalla “Motor” encontramos los valores nominales del motor, además del tipo de macro utilizada por el convertidor para el control. Es una pantalla de solo consulta.



Figura 164 - Panel Builder: Programa control y supervisión HMI 3

En la pantalla “Variador” accedemos a la visualización de los parámetros del convertidor ACS550. Es una pantalla de supervisión del sistema y, por tanto, no existen variables de modificación. También, se presenta el ultimo error producido en el variador.



Figura 165 - Panel Builder: Programa control y supervisión HMI 4

Finalmente, en la pantalla configuración, se dispone de tres pantallas para modificar y configurar el convertidor ACS550. Por motivos de seguridad, se ha incluido un botón de aplicar, con lo cual, solo se aplicará el cambio si se presiona dicho botón. A continuación, se presentan las tres pantallas de configuración.



Figura 166 - Panel Builder: Programa control y supervisión HMI 5



Figura 167 - Panel Builder: Programa control y supervisión HMI 6



Figura 168 - Panel Builder: Programa control y supervisión HMI 7

Todos los parámetros modificables, incluida la velocidad, están limitados a los valores máximos y mínimos del convertidor ACS550. Por lo tanto, no se pueden introducir consignas que puedan perjudicar la seguridad del sistema. Los parámetros están configurados, de manera inicial, con los valores por defecto del convertidor.

10. Conclusiones

Una vez finalizado el estudio, se considera que el alcance del proyecto se ha desarrollado con éxito, y se concluye lo siguiente:

- ABB dispone de gran variedad de documentación y formación, para iniciarte en sus herramientas de manera rápida y sencilla
- El sistema es robusto y confiable. Los dispositivos de ABB empleados, como los autómatas AC500, el convertidor ACS550 y los paneles CP600 muestran gran calidad y un alto rendimiento.
- El software Automation Builder es una herramienta potente. Las posibilidades de integración de todos los dispositivos del sistema de automatización facilitan, de forma clara, todo el proceso. Además, el hecho de estar integrado todo el sistema permite una reducción de tiempos por eliminar las reprogramaciones innecesarias.
- El software Panel Builder facilita el trabajo de programación con un entorno gráfico e intuitivo. A la vez que dispone de herramientas para la creación de sistemas de última generación.
- El software CODESYS, respaldado en la norma IEC61131-3, ofrece unas prestaciones técnicas altas, versatilidad y flexibilidad. El conocimiento de los distintos lenguajes permite un potencial de programación que se adapta a las necesidades actuales, indiferentemente, del fabricante de los dispositivos.
- Los softwares utilizados simplifican el trabajo de comunicación al utilizar los distintos equipos de ABB a través de las bibliotecas existentes.
- El sistema de control realizado es confiable y eficiente. La interfaz gráfica HMI es eficaz y visualmente atractiva, organizada de manera que minimiza la posibilidad de error.
- Se ha dispuesto de una guía clara y concisa para el desarrollo de proyectos futuros basados en la tecnología ABB.

11. Bibliografia

- [1] ABB Library. (2015). *Plataforma de automatización AC500. Herramienta de ingeniería Automation Builder 1.2*. Recuperado de <https://library.abb.com/en>
- [2] Zhañay, O. (2015). *Estudio del sistema de adquisición y monitoreo de temperatura de las unidades de generación de la central Paute Molino y elaboración del manual de operaciones* (Trabajo fin de carrera). Universidad Politécnica Salesiana, Cuenca.
- [3] ABB Library. (2016). *ABB Automation Builder 2.0. Next Level Engineering Productivity*. Recuperado de <https://library.abb.com/en>
- [4] ABB Library. (2018). *Automation Builder Engineering Suite. Installation and Activation*. Recuperado de <https://library.abb.com/en>
- [5] ABB Library. (2011). *Application guide. ACS355 and AC500-eCo*. Recuperado de <https://library.abb.com/en>
- [6] ABB Library. (2012). *Quick start Guide. ABB PLC and drives integration using Modbus RTU*. Recuperado de <https://library.abb.com/en>
- [7] ABB Library. (2011). *Application Note. AC500 PLC and ABB ACS355 Drive via Modbus RTU with ABB Standard Library*. Recuperado de <https://library.abb.com/en>
- [8] Grupo de Investigación SUPRESS. (2008). *Introducción al software de programación CODESYS versión 1.0*. Recuperado de <http://lra.unileon.es/>
- [9] Lorenzo, N. (2006). *Aplicación docente de una plataforma de accionamientos mecatrónicos controlada a través de Internet*. (Proyecto fin de carrera). ETSEIB. Recuperado de <https://upcommons.upc.edu/handle/2099.1/3620>
- [10] Mateos, F. (2008). *Autómatas Programables: Introducción al Estándar IEC-61131*. (presentación). Universidad de Oviedo. Recuperado de <http://www.isa.uniovi.es/genia/>
- [11] ABB Library. (2011). *Plataforma de automatización escalable AC500. Direccionamiento IP*. Recuperado de <https://library.abb.com/en>
- [12] ABB Library. (2011). *Operating instruction. Control Panels CP600*. Recuperado de <https://library.abb.com/en>

- [13] ABB Library. (2014). *Operating Instructions. PB610 Panel Builder 600. Programming Software for CP600 Control Panels*. Recuperado de <https://library.abb.com/en>
- [14] ABB Library. (2018). *Installation instructions CP600*. Recuperado de <https://library.abb.com/en>
- [15] ABB Library. (2018). *Data sheet Control Panel CP600*. Recuperado de <https://library.abb.com/en>
- [16] ABB Library. (2011). *Plataforma de automatización escalable AC500. Modbus RTU*. Recuperado de <https://library.abb.com/en>
- [17] ABB Library. (2010). *Plataforma de automatización escalable AC500. Ethernet Modbus TCP*. Recuperado de <https://library.abb.com/en>
- [18] ABB Library. (2010). *Quick start Guide. ABB PLC and drives integration using Modbus RTU*. Recuperado de <https://library.abb.com/en>
- [19] ABB Library. (2018). *Data sheet TB511*. Recuperado de <https://library.abb.com/en>
- [20] ABB Library. (2018). *Installation instructions CD522*. Recuperado de <https://library.abb.com/en>
- [21] ABB Library. (2018). *Assembly manual AC500 PLC. System Assembly and Device Specifications for AC500 V2 Products*. Recuperado de <https://library.abb.com/en>
- [22] National Instruments. (2019). *Información Detallada sobre el Protocolo Modbus*. Recuperado de <http://www.ni.com/es-es/innovations/white-papers/14/the-modbus-protocol-in-depth.html>
- [23] Bentz. (2019). *Protocolo Modbus*. Recuperado de: <http://bentz5.blogspot.com/2015/10/protocolo-modbus.html>
- [24] ABB Library. (2018). *INSTALLATION INSTRUCTIONS AC500 Product Family*. Recuperado de <https://library.abb.com/en>
- [25] ABB Library. (2018). *Installation instructions DA501*. Recuperado de <https://library.abb.com/en>

- [26] ABB Library. (2018). *Automation products AC500, CP400, CP600, DigiVis 500, Wireless*. Recuperado de <https://library.abb.com/en>
- [27] ABB Library. (2018). *AC500 – The scalable PLC for customized automation*. Recuperado de <https://library.abb.com/en>
- [28] ABB Library. (2018). *Installation instructions PM573-ETH*. Recuperado de <https://library.abb.com/en>
- [29] ABB Library. (2018). *Data sheet PM573-ETH*. Recuperado de <https://library.abb.com/en>
- [30] ABB Library. (2018). *Manual del Usuario. Convertidores de frecuencia ACS550-01 (0,75...160 kW). Convertidores de frecuencia ACS550-U1 (1...200 CV)*. Recuperado de <https://library.abb.com/en>
- [31] Puigneró, P. (2014). *Aplicación de convertidores de frecuencia en la regulación del motor asíncrono y estudio de las interferencias electromagnéticas generadas*. (Proyecto fin de carrera). Universitat Politècnica de Catalunya, Barcelona.
- [32] Piñero, J. (2015). *Control de un motor de inducción usando un variador de frecuencia*. (Proyecto fin de carrera). Universidad de Sevilla.